PRIJS ƒ 7.25/Bfr. 135

Commodore INFO

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS

JAARGANG 5, NO. 7, november 1988



Commodore Info

Verschijnt 8x per jaar

Jaarg.5, no.7, nov. 1988

Uitgave:
Sala Communications/SAC
Uitgever:
Vic Sharfman
Redactie:
Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdredacteur
drs. J. Boers eindredakteur
drs. M. de Rooij, drs. H. Zoete,
H. Smeenk, drs. U. Schuurmans,
R. Goudriaan, B. Munniksma,
P.C. Broekhuizen.
Redactiesecretariaat:
R. van Zalingen
Tel. 020-228871
Strip: Bert Tier
Illustraties: Ben van Mierlo
Advertentie-exploitatie:
Ing. V. Sala, Ing. B. Sala,
D. van Vlijmen
Weesperstraat 103
1018 VN Amsterdam
tel. 020-273198
Redactie adres:
Postbus 43048
1009 ZA Amsterdam
tel. 020-228871
Listingtelefoon:
(ma: 17.00-21.00) 02155-25162
Abonnementen en administratie:
Nicole Balke en Marjo Jansen
Postbus 43048
1009 ZA Amsterdam
tel. 020-248006
Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.
Abonnement:
Voor 8 nummers ƒ 47,50 of Bfr. 975 per jaar. Betaling op giro 1585491 t.n.v. SAC/Commodore-Info. Oude nummers kunt U alleen krijgen bij vooruitbetaling van ƒ 6,75 op de bovenstaande rekening. Ook telefonische opgave voor een abonnement is mogelijk. Bel GRATIS 06-02242222 (teleservice), elke dag tot 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze telefoonnummers zijn alleen bedoeld voor opgave van NIEUWE abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.

Druk: NDB, Zoeterwoude
Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

ISSN: 0169-3085

Inhoud van dit nummer

Coverfoto:
Lights, Camera, Action ! (Aegis)
(met dank aan Altycos Zoetermeer)

Redactioneel

Op de Efficiency Beurs was Commodore duidelijk als zakelijke leverancier aanwezig, men wil PC's en Amiga's leveren aan het bedrijfsleven. Dat is begrijpelijk, er zit nog een grote groei in de computermarkt daar. Het is echter jammer, dat men dan wat minder tijd over heeft voor het huiscomputersegment. Mooie plannen zijn fraai, in de praktijk heeft men in de Commodore organisatie minder 'liefde' voor haar oorsprong en de pijlers als de 64 dan we mogen verwachten. Praten over nieuwe marketing plannen is mooi, maar de huidige klanten mogen niet in de kou komen te staan. We vragen echt zeer geregeld of er nog iets nieuws is voor de 64 of de 128, om over de bijna vergeten Plus 4 en C-16 (en Vic 20) maar niet te praten. Maar helaas, veel is er niet te melden. Voor de Amiga gaat het wat beter, de nieuwe harde schijven, de koppeling aan laserprinters en kleurenplotters, de hele Unix overstap, daar gebeurt nogal wat. Ondertussen blijken onze collega's van Dossie Commodore het bijltje er bij neer gegooid te hebben, dat blad houdt, naar verluidt, op te bestaan. We zijn als Commodore Info dus weer alleen op de markt, maar we blijven u bedienen. En we zien zeer velen van u vast weer op de Commodore Info Beurs in de RAI op 11 en 12 november, waar u als vanouds natuurlijk weer veel C-64, C-128 en Amiga spullen kunt vinden. De PC is er ook, maar dat begrijpt u ook wel.

Luc Sala

De Orgatechnik tentoonstelling in Keulen is vlak na de Efficiency Beurs toch een voor ons land belangrijk gebeuren. Heel wat Nederlandse bedrijven gebruiken die beurs om onze Oosterburen te bedienen en met name op Commodore gebied was het een goede gelegenheid weer eens up- to-date te komen.

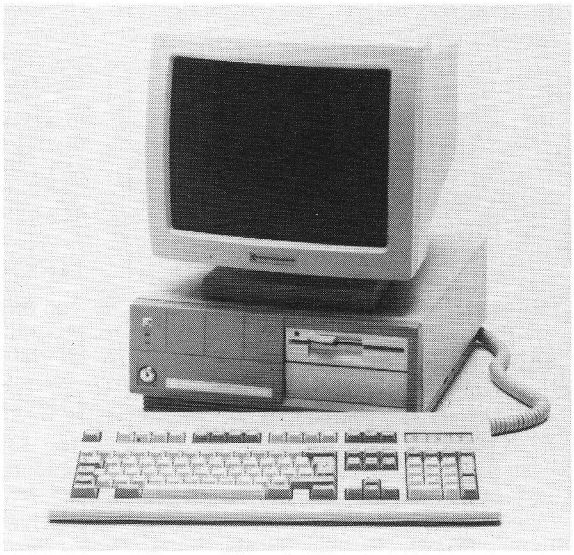
Orgatechnik Keulen: traditionele beurs

Het samenbrengen van kantoorinrichters, computerhandel en kopieermachines op een beurs leidt tot onoverzichtelijke massa-toestanden. Net als de EB was de Organtechnik dan ook veel te groot, veel te weinig geconcentreerd en tamelijk warrig. Druk was het er wel en de sfeer is er toch anders, in Duitsland is de binding klant-leverancier veel sterker.

Op zo'n beurs als de Orgatechnik komt de klant vooral om bestaande banden met leveranciers te onderhouden, om in een persoonlijk gesprek te horen wat nieuw is en daarvoor dan ook een opdracht te geven. Er worden dus direct zaken gedaan, de orderboeken liggen altijd klaar en de stemming is wat serieuzer dan op de EB. Er is geen tijd voor stunts en het trekken van publiek is ondergeschikt aan het demonstreren en verkopen. Bij onze Oosterburen krijgt men niet snel een goede reputatie, maar als die er eenmaal is, blijft men ook zaken doen.

Traditie en trouw

Bedrijven met een lange traditie en goede naam zoals Philips doen het dan ook heel goed, men levert al jaren aan bepaalde klanten en die lopen echt niet voor een beursaanbieding naar de concurrent. Men wil wel nieuwe spullen, maar dan wel goed uitontwikkeld, tijd voor kinderziekten heeft men niet. De prijzen zijn relatief hoog, maar men verwacht wel service en die moet dan van de dealer komen, het land is te groot voor een enkele servicedienst. Op de stands van de diverse leveranciers is er dan ook altijd veel plaats voor die dealers en softwarepartners. Die krijgen de kans, hun toepassingsprogramma's en verticale (branche)-pakketten te demonstreren en te verkopen, zo realiseert men de omzet in hardware. Zelfs een bedrijf als IBM verkoopt bijna 90% van haar PC omzet via de dealers, de directe verkoop is zeer ondergeschikt. IBM kent in Duitsland ook geen distributeurs, men wil direct contact houden met de dealers, die zijn daar te belangrijk voor. De recente ophef rond het afschaffen van de 'distributors'-functie in ons land en in Frankrijk kwam dan ook in Duitsland niet voor.



De nieuwe PC 40-III van Commodore

Commodore

In Duitsland is het marktaandeel van Commodore Buromachinen GmbH nog steeds heel groot, men is er duidelijk een A-merk. De jarenlange relatie met de dealers en softwarehuizen is overeind gebleven in de overgang van 8 bit naar 16 bits machines en Commodore is daar dan ook duidelijk op de PC toer gegaan. De Amiga wordt goed verkocht als hobbymachine, maar voor zakelijk gebruik gaat het nagenoeg helemaal om de PC en AT lijn. Die worden gezien als Duitse machines, ze worden dan ook in Braunschweig gemaakt en veel Duitsers weten niet dat Commodore een Amerikaans bedrijf is.

Aandacht

De Duitse pers geeft dan ook veel aandacht aan het bedrijf en op de wat statige persconferentie in Keulen zaten er dan ook wel een paar honderd man pers in de zaal. Niet dat de boodschap van Harold Speyer, de Duitse topman, nu zo inspirerend is, maar het is gewoon een verplicht nummertje voor de journalisten daar. De wat droge cijfers en als altijd optimistische verwachtingen worden als zoete koek geslikt. Omdat Duitsland langzamerhand de kurk is, waarop het hele Commodore gebeuren drijft, is het duitse management-team oppermachtig en ziet men vrijwel niemand uit de VS. De Duitse omzet is dan ook met 866 miljoen DM bijna 55% van de totaalomzet van Commodore wereldwijd. Het marktaandeel van het bedrijf in Duitsland is met 70.000 verkochte PC's het afgelopen jaar weer iets gegroeid, men is nu duidelijk nummer 2 na IBM.

Komplett-Netz

In ons land ging men op de EB al fors in de richting van een Unix netwerk met Commodore hardware, op de Orgatechnik hield men het bij de iets traditionelere Lokale Netwerken met de PC 60 als fileserver met een 80386 processor. Een klein netwerkje voor gebruik in kleinere bedrijven, dat Anwender Netz heet, wordt kompleet geleverd, met alle kaarten en software al geinstalleerd. Met een PC 60 en twee

PC 10- III's, Netboard kaarten en het Network OS plus het geïntegreerde softwarepakket Superdesk in netwerkversie is dat een handige oplossing voor het MKB. Het is niet erg snel, de transmissiesnelheid is 2,5 Megabit/sec, met datablocks van 48 KB. Maar er wordt dan ook gebruik gemaakt van relatief eenvoudige (RS 422)-bekabeling, niet van coaxiaalkabel. Er kunnen tot maximaal 10 PC worden aangesloten op het 'Anwender-netz', de maximale kabellengte is 150 meter. De Net-Bios maakt het netwerk compatibel met software, die voor Novell geschreven is.

Computhek

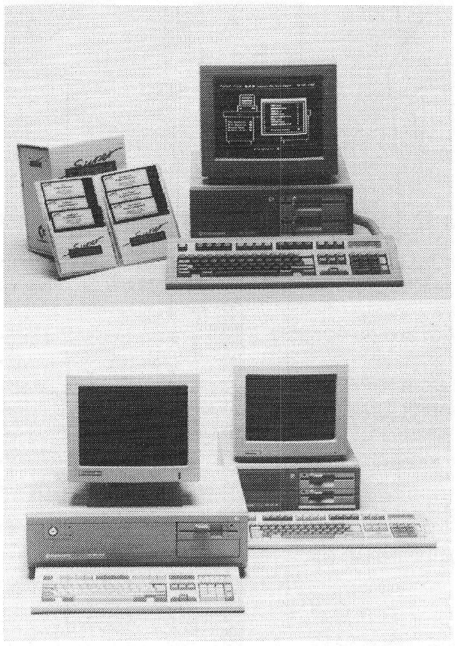
Om de kennismaking met de computer en het op de hoogte blijven van de verschillende software-toepassingen eenvoudiger te maken, heeft Commodore het 'Computhek-Programm' gemaakt. Dat is een leerprogramma, dat ook de kans geeft om interactief verschillende soorten software te verkennen. Het geheel is in leerfasen opgebouwd, waarbij men ook als beginner al kennis kan nemen van wat 1-2-3, dBase of Wordstar te bieden hebben. Wanneer men het Computhek programma voor bedrijfstrainingen gebruikt, kunnen de uit te leggen softwarepakketten naar behoefte worden aangevuld.

PC 60

De eisen, die tegenwoordig aan een 80386 machine gesteld worden, worden steeds hoger. Met name omdat OS/2 en de diverse Unix varianten een veel groter geheugen dan de vertrouwde 640 KB nodig hebben, is de geheugentoegang van belang. Die dient 32 bit te zijn en omdat er op dit moment nog geen standaard is (m.u.v. IBM's MCA en de toekomstige EISA) zet men dan liefst zo veel mogelijk RAM op de moederkaart. Daar kan men wel via een 32 bits datapad gegevens van en naar het geheugen krijgen. Bij de PC 60 van Commodore kan men tot 2,5 MB op die manier benutten, waarbij gebruik gemaakt wordt van de 'Control/386' software van Phoenix Technologies. Daarmee wordt zowel de indeling van de onderste 720 KB, het gebruik van het RAM volgens EEMS en LIM, maar ook cache geheugen voor disk-optimalisatie geregeld.

PC 40-III

Van de meeste Commodore PC's komen er geregeld nieuwe modellen. De PC- 40 is aan versie III toe, die vooral qua uiterlijk heel wat kleiner geworden is. Meer onderdelen en interfaces op het moederboard, dat is de oplossing die men hier koos. Naast de videocontroller (met ook VGA) zijn ook andere I/O interfaces nu standaard op de kaart gezet.
De kloksnelheid is ook verhoogd, men kan met 12 Mhz draaien, maar ook terugschakelen, hetgeen voor bepaalde spelletjes nuttig kan zijn. Er zijn vier vrije sleuven in de PC 40-III.



Anwender-Netz met PC 60

Amiga

Commodore had in Keulen duidelijk meer aandacht voor de zakelijke bezoekers dan voor de creatieve Amiga uitspattingen. Met name zochten we tevergeefs naar de aangekondigde Workbench 1.3, (zie de bespreking elders van de onofficiele versie) maar gelukkig brachten de wel aanwezige Amiga toeleveranciers op de beurs ons weer helemaal up-to-date.
Volgens het bedrijf staan er 200.000 Amiga's in Duitsland. Dat is een fors aantal en er is dan ook een levendige handel in Amiga spullen.

Universeel scanner

Ludwig uit Munchen (tel. 089-3113066) levert de Silver Reed universele scanner, in een speciale Amiga uitvoering voor 1498 mark. Dat is niet veel geld voor een A-4 scanner, compleet met ingebouwde thermische afdrukker en scan-software. De resolutie is met 200 dpi wat mager, maar doordat men 16 grijstinten (4 bits) in IFF formaat kan onderscheiden, is het resultaat acceptabel. e twijfelen zelfs aan die 200 dpi, de scherpte in één richting is duidelijk beter dan in de andere. De afdrukkwaliteit valt tegen, men kan beter de zaak inlezen en uitdraaien met een echte laserprinter, maar de thermoprinter is dan ook niet meer dan een soort proefdrukker. Het inlezen van een A4 afbeelding duurde ongeveer 15 seconden, men kan ook verkleinen tot max. 1:4.
Er is wel een Amiga met minstens 1 MB aan geheugen nodig, gescande plaatjes vreten nu eenmaal bytes aan opslagruimte. De scanner komt kompleet met interface, kabel, software en handleiding. Qua prijs heel redelijk, kwalitatief hebben we wel eens betere scanners gezien, die dan ook, moeten we toegeven, heel wat meer kosten en geen print- functie kennen.

Software-aanbod

Bij software-uitgeverij Markt&Technik bekeken we nog even hun winter catalogus. De interessante 'DeLuxe Productions' voor de Amiga van EA is helaas alleen in de NTSC versie leverbaar (we hebben hier PAL nodig). De andere EA DeLuxe programma's, zoals Print II en Photolab Amiga deden ons toch weer even stilstaan, de creatieve mogelijkheden van de Amiga zijn toch steeds weer verrassend.
Superbase Professional is een uitstekend database programma voor de Amiga, met een goede naam uit de C64 en zelfs de 8032/Pet historie. Helemaal aan te passen aan de meeste vreemde database wensen is het van belang, dat dit programma ook draait op een 512 KB Amiga.
Voor de C-64 en de 128 zijn er ook wat leuke nieuwe programma's. In Duitsland biedt men ook de duitse versie van GEOS128 aan (DM 119) en de toepassingsprogramma's GeoCalc en Geofile voor de 128.

Doelgoep: meisjes

Voor de Amiga 500 heeft men een speciale aktie bedacht, die zich richt op meisjes. Het thema is de computer als vriendelijke helper bij het huiswerk. Men levert daarom het Amiga 500 Power Pack, compleet met allerlei software. Als motto gebruikt men : 'Kiss me, Amiga' en er is een Miss Amiga verkiezing.
In Duitsland is er ook een speciaal onderwijs model van de Amiga, de 2000E. Daarvan zijn er nu meer dan 5000 verkocht.

L.S.

Op de Efficiency Beurs deed Commodore Nederland haar marktstrategie uit de doeken. Daarbij speelt de Amiga een dubbelrol, als (re)creatieve computer, maar ook als professioneel workstation.

Commodore wil naar zakelijke markt

Amiga en PC zij aan zij in C.O.M.E. concept.

Op de Commodore stand van de Efficiency Beurs was nauwelijks meer een leuke toepassing te zien, alles was bloedserieus, de nieuwe zakelijkheid van CBM voerde de boventoon. Teleurstelling voor de 64 gebruiker? Ach, zo'n EB is toch echt een zakelijk gebeuren geworden, de huiscomputer past weer meer op andere beurzen.



Links: J.F. van Lemmen Commercial Director

Rechts: B.P.H. van Tienen Managing Director

Koppeling, compatibiliteit en samenwerking met grotere computers, dat interesseert op dit moment het bedrijfsleven. Dat staat vol met PC's, maar kijkt uit naar de volgende stap. Met de basis van een stevige range MS-DOS compatibele computers, een steeds krachtiger Amiga familie en met de blik duidelijk naar de Unix toekomst wil Commodore een krachtiger positie bij de zakelijke kopers bereiken.

Ommezwaai

Op dit moment is Commodore nog voor 80% van haar omzet afhankelijk van de consumentensector, de huiscomputers. Men wil dat gaan veranderen en vooral in de professionele sfeer gaan groeien. De verhouding moet 40/60 gaan worden en dat wil men met name gaan bereiken, door de meerwaarde van het zakelijk aanbod te verhogen. Meer support, betere kwaliteit en vooral een zodanige samenwerking met partners in de markt, dat men de klant bredere oplossingen kan bieden. Alleen PC's leveren kan niet meer, er is behoefte aan systemen, die aan elkaar en aan andere systemen gekoppeld kunnen worden, ook naar grotere netwerken en minicomputers toe, vindt de heer van Lemmen. Toegevoegde waarde, wil men de stap naar A-merk kunnen waarmaken, dan zal men de klanten meer moeten bieden dan de doos met het apparaat.

C.O.M.E

Om de zakelijke gebruikers dat brede pakket te kunnen bieden, heeft Commodore samen met het bedrijf Matricom het COME concept opgezet. Dat is het Commodore Open Management Environment, een raamwerk voor het inpassen van Commodore hardware in de meest voorkomende netwerken. Commodore ziet zichzelf daarbij als de naaf in een netwerk van partners, die bijvoorbeeld hun netwerk software, zoals Banyan of Novell, operating systeem zoals SCO Xenix/UNIX of communicatie-faciliteiten toevoegen. Maar ook koppeling van andere randapparatuur, communicatie naar mini-computers zoals die van Digital en Pyramid via Ethernet en het gebruik van uitbreidingskaarten van Add-On. Samenwerken en dan wordt het beeld van het 'Winning Wheel', met Commodore in het hart, gebruikt om de samenwerkingsvorm aan te geven. Haar partners heten dan VAR ofwel value Added resellers, een Amerikaanse kreet.

Unix-trend

Hier valt een duidelijke trend naar Unix te merken, het bedrijf bereidt zich duidelijk voor op een offensief in die richting met de nieuwe Amiga 2500 UX met 68020 processor en de 3000 Amiga, die volgend jaar uitkomt met een 68030 hart. Naast de Amiga's voor de zakelijke toepassingen blijft natuurlijk ook de PC en AT lijn van Commodore een krachtig bestanddeel van het COME plan. Op de Efficiency Beurs liet men de COME implementatie zien, door in een nagebouwd kantoor de totale koppeling van alle Commodore systemen en partner-aanvullingen tot één geïntegreerde systeem te realiseren. Met minicomputers, snelle Ethernet-LAN's, de laatste snufjes waren daa-

rin samengebracht. Doordat men in zo'n omgeving toch kan blijven werken met dezelfde software, bijvoorbeeld het op de Amiga, onder MS-DOS en onder Unix draaiende Word-Perfect, wordt het voor de gebruiker een bereikbaar systeem en voor de beheerder een overzichtelijke en toekomstzekere aankoop.

Duitsland als voorbeeld

In het buurland is het Commodore gelukt, op te klimmen tot de tweede plaats als leverancier van PC's aan het bedrijfsleven. Men heeft daar het voordeel van eigen ontwikkeling en produktie, vlak bij Hannover, maar het succes van de Westduitse organisatie is ook wereldwijd een van CBM's belangrijkste troeven. Men is daar nooit de positie als gewaardeerde leverancier aan tientallen softwarehuizen en VAR's, begonnen met de PET en 8032 tot 8096 systemen, kwijtgeraakt. Geleidelijk zijn die bedrijven overgeschakeld op PC's en de Duitse organisatie heeft vanaf het begin gezorgd dat men de daarvoor benodigde hardware, met name de PC lijn, kon leveren. Zelfs waar men in de VS niet aan de PC wilde, werd die in Duitsland ontwikkeld en naar nu bleek, terecht, want het marktaandeel daar is indrukwekkend.

Terugwinnen

In ons land, waar zo'n drie tot vijf jaar geleden een enigszins vergelijkbare situatie bestond, met een duidelijke positie voor CBM in de professionele automatisering van het MKB en in bepaalde verticale markten, is dat minder goed gelukt. De ooit zo aan Commodore verknochte dealers zijn in het huiscomputer-geweld wat vergeten, en de PC's zijn wel breed, maar niet erg gericht afgezet. De huidige Commodore PC dealer of wederverkoper moet het hebben van prijs en algemene naamsbekendheid, de verticale branche-oriëntatie van weleer is goeddeels verdwenen. Commodore erkent ook, dat men met het PC aanbod eerder marktvolger is dan dat men zelf met nieuwe aanpassingen komt. Indien de markt er om vraagt, komen wij met bijvoorbeeld de VGA beeldstandaard, zoals in de nieuwe AT's, maar niet eerder. Daarom verwachten we ook zeker, dat Commodore een EISA machine (de nieuwe busstandaard als alternatief voor IBM's MCA) gaat leveren, wanneer de markt daarom vraagt.

Men ziet op dit moment meer in een doordachte koppeling en integratie van de huidige techniek, voorop lopen in toepasbaarheid, niet in technologische hoogstandjes. Dat is waarom men ook met COME wil komen, en vooral de grotere accounts overtuigen van de waarde daarvan door een doordacht en vooral qua communicatie breed concept. Die grotere klanten worden via telemarketing op de hoogte gebracht van het COME plan en men wil daarmee de directe relatie met grotere klanten verbeteren.



Commodore Amiga 2000

Amiga: veel gezichten

De rol van de Amiga wordt steeds belangrijker, omdat men nu ook Amiga's heeft, die geschikt zijn voor serieuze toepassingen. De Amiga 2000, met de PC emulatie-opties, was al een goede stap in die richting, maar er komt nog meer. Dat is in de eerste plaats de Amiga 2500, die ook onder Unix kan draaien. Met een prijskaartje, dat tegen de 15.000 gulden gaat liggen, is dit duidelijk geen hobbymachine meer, maar een volwaardig werkstation of zelfs de kern van een multi-user Unix systeem. Begin volgend jaar komt er ook een model 3000, dat nog krachtiger zal zijn door het gebruik van de Motorola 68030 processor. Voor gebruik in netwerken zijn er nu diverse oplossingen voor de Amiga. Zo zijn er interface kaarten naar de typische PC netwerken, zoals Arcnet, die ongeveer 1000 gulden gaan kosten. Wil men echter een compleet opgetuigde Amiga gaan gebruiken als Unix werkstation, dan kan er ook een directe Ethernet koppeling (met TCP/IP protocol) worden aangeschaft. Zo'n kaart kost dan wel tegen de 3000 gulden, maar het gaat dan om geavanceerde CAD/CAM of video-toepassingen, waar de creatieve vrijheid van de Amiga die van vergelijkbare PC constellaties duidelijk overtreft.

Thuiscomputeren

Bij alle plannen voor een actief marketingbeleid op de zakelijke markt zal men de bread-and-butter lijn, de afzet van huiscomputers, niet laten lopen.

Volgens van Tienen heeft Commodore op de markt voor huiscomputers een riante positie. Wereldwijd zijn er 750.000 Amiga's verkocht. Zeker nu MSX min of meer wordt afgebouwd, alleen Philips verkoopt het maar in een paar landen, heeft men het in het segment tot ongeveer 500 gulden voor het zeggen. De C-64 en C-128 zijn daar nog sterke produkten, al is de leeftijd van de gebruikers wel wat jonger geworden. Men geeft toe, dat er een zekere stabilisering of zelfs inzakken van de afzet is opgetreden, daarentegen zijn de prijsoorlogen op deze markt verleden tijd. Met de Amiga 500, waarvan er tegen het einde van het jaar in ons land zo'n 30.000 zullen zijn verkocht, heeft men een sterke positie in het segment van de creatieve huiscomputers, met alleen de Atari ST als concurrent. Commodore gelooft echter, dat men die strijd glansrijk gewonnen heeft en dat de Amiga verder zal doorgroeien tot de 'standaard' huiscomputer voor wie meer wil dan wat de toch vrij beperkte budget-PC-kloon kan bieden.

Spelcomputer

In Japan en later in de VS heeft Nintendo bewezen, dat er wel degelijk belangstelling is voor pure spelcomputers, in de traditie van de Atari modellen uit het begin van de jaren tachtig. Die markt is in de VS weer gaan opbloeien en met name voor jongere kin-

deren worden er weer miljoenen video-spelmachines verkocht. Atari is ook weer actiever geworden en heeft nieuwe spelcomputers uitgebracht, en dan kan Commodore niet uitblijven. Op de Consumer Electronics Show in Las Vegas begin januari volgend jaar komt er dan ook een 16 bits Commodore spelcomputer uit. Men wil nog niet antwoorden op de vraag, of die ook compatibel zal zijn met de C-64. Wanneer dat het geval is, dan komt de 'oude' C-64 software, die vroeger ook vaak op insteek-cartridges werd gezet, weer in beeld. Maar dat houdt ook in, dat voor de nieuwe machine al een berg software min of meer van de plank gehaald kan worden. Dat wil Commdore misschien liever niet, men wil liever de sofwtarebedrijven nieuwe software zien ontwikkelen. Of dat gebeurt, is de vraag, Commodore heeft wat betreft de software voor haar systemen nog steeds een vreemde opstelling.

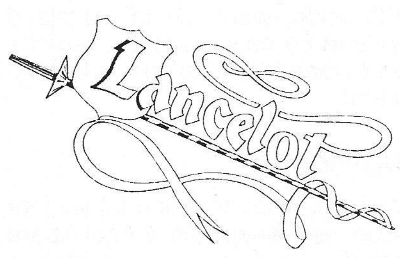
De software Januskop

Voor de professionele en zakelijke markt is men absoluut tegen kopiëren en ziet men het belang van een gezonde softwarebranche in. Voor de thuiscomputermarkt daarentegen is de opstelling veel vrijer. Daar kopieert men er toch naar hartelust op los, dat kan ons niet zoveel schelen, zegt men. Dat is daar deel van de sport en we gaan daar ook in mee door bijvoorbeeld voor de Amiga steeds meer software met het systeem mee te verkopen. De software is een soort Freeware, maar daarbij vergeet men wel, dat de ontwikkeling van betere en nieuwe software op die manier stagneert. Zeker in ons land, dat wereldwijd een heel slechte naam heeft als software-piraterijnest, merken we dat heel duidelijk. De weinige bedrijven in ons land, die zelfstandig software konden ontwikkelen voor de huiscomputermarkt, zijn ofwel failliet (Aackosoft) ofwel overgestapt op serieuze PC software zoals Radarsoft. De verkrijgbaarheid van Amiga software is bedroevend slecht, omdat iedere importeur en detaillist weet, dat het spul toch snel gekraakt wordt en het kopieercircuit in gaat. Men wil niet adverteren voor nieuwe software en het is ook duidelijk, dat bijvoorbeeld voor de C-64 de ontwikkeling ervan ook bijna is gestopt. Niemand gelooft er meer in, voor de tijdschriften die deze markt toch voor een deel aan de gang houden, is er weinig meer om over te schrijven.
Commodore zou er daarom beter aan doen, wat betreft het kopiëren één lijn te trekken, en op te treden tegen deze praktijken. Men kan niet voor de ene soort software tegen kraken zijn en voor de andere soort dit toejuichen.

Luc Sala

Koning Arthur in de computer

De sage van de Heilige Graal, Lancelot en Koning Arthur wordt weer levend. Ditmaal in een computer-avonturenspel en natuurlijk is de Amiga het uitnemende platform om zo'n spel gestalte te geven. Het is een semi-klassiek, voor zover we daarbij in het geval van de Amiga kunnen spreken, beeld-adventure geworden. Om het nog aantrekkelijker te maken heeft de uitgever van het het spel, Mandarin Software in Groot Brittannië er nog een prijsvraag aan verbonden die de winnaar een kostbare kist met zilveren graal oplevert van 5000 pond sterling. Beproef je geluk, als je tot 2 februari 1989 naar de 'Hotline' in Engeland belt en de geheime bergplaats van de graal weet op te sporen.



Bij dit spel wordt gebruik gemaakt van de 'parser' van het bekende level 9. Een parser is een stuk software, dat de link legt tussen gewone (in het engels dan wel) commando's en wat er in het spel gebeurt, een soort vertaler dus. Deze parser is zeer befaamd en dat men er mee mag werken verhoogt de spelkwaliteit aanzienlijk. Gebruik van commando's als GO TO en RUN TO maakt reizen door het adventure veel eenvoudiger. En UNDO is de redding voor wie te snel sneuvelt in de speurtocht naar de Graal.

Ronde tafel ridders

Het spel heet 'Lancelot', is ontwikkeld door de 'Level 9'-groep en bevat drie avonturenspelen met een totaal van meer dan 60.000 woorden tekst, afgewisseld door fraaie plaatjes. Het spel volgt nauwgezet het boek 'Le Morte d'Arthur' van Sir Thomas Malory, zoals dat in 1485 door William Caxton gepubliceerd werd. Gelukkig wordt niet de Middeleeuwse tekst gebruikt voor het spel, maar ademt het wel de magisch-sacrale sfeer van het origineel.

Historisch

Het begint uiteraard bij het stichten van de Tafelronde en eindigt bij de succesvolle jacht (in het middelnederlands 'queeste') op de Heilige Graal.
Heel smaakvol is bij het spel achtergrond-informatie en een perkamenten kaart van Engeland in Koning Arthurs tijd bijgevoegd.
De hele Saga bestaat uit drie in elkaar verwikkelde adventures, allemaal gelardeerd met puzzles, raadsels en maar liefst 60.000 woorden tekst. Geen opgave om in een uurtje te doen, hier zit weken spelplezier in, het electronische avonturenboek in optima forma.
Lancelot wordt uitgebracht in verschillende versies: Atari ST, Commodore Amiga, Commodore 64, Apple Macintosh, PC's en compatibelen, Amstrad CPC/PCW, BBC Master, Spectrum Plus3, in deze formaten met graphics. In tekstvorm ook nog voor de Apple II, de Atari XL/XE, de Sinclair Spectrum en de MSX.
Natuurlijk zijn het vooral de Amiga en ST versies, die het qua screenshots zo heel mooi doen en eigenlijk de andere versies wat in de schaduw stellen.

Mandarin Software, Europa House, Adlington Park, Adlington, Macclesfield SK104NP, U.K. Deze Amiga software wordt echter ook door Computer Collectief en gespecialiseerde dealers verkocht.

De serie Basic-artikelen wordt door Jan Bodzinga voortgezet met een reeks afleveringen, waarin alle Basic-commando's en functies aan de orde komen, die voor de Commodore computer te gebruiken zijn. Na de algemene introductie in de vorige les, beginnen we nu met het echte werk. De cursus is bewust niet gerelateerd aan een bepaald type uit de Commodore reeks, maar kan in principe bij iedere computer met Basic interpreter worden toegepast.

Basis Basic

deel 24: Basic van A tot Z

De Basic interpreter van de Commodore kent een hele reeks opdrachten en functies. In de vorige aflevering zijn we uitvoerig ingegaan op de manier waarop we de computer duidelijk kunnen maken hoe we een opdracht precies uitgevoerd willen hebben, terwijl natuurlijk ook de parameters bij de functies aan de orde zijn gekomen.

Deze les zullen we onze excursie door het Basicland beginnen met de opdrachten die alfabetisch gezien bovenaan staan. We hebben bewust voor deze opzet gekozen om een zo breed mogelijk scala door elkaar te kunnen behandelen, zodat het geheel wellicht nog leesbaar blijft, terwijl ook het terugzoeken van een bepaalde functie achteraf tamelijk snel in z'n werk kan gaan. Een nadeel van deze opzet is natuurlijk dat het nog wel even duurt voor we opdrachten als WAIT en VERIFY zullen behandelen. Maar uiteindelijk zullen deze het geheel completeren.
Laten we maar beginnen:

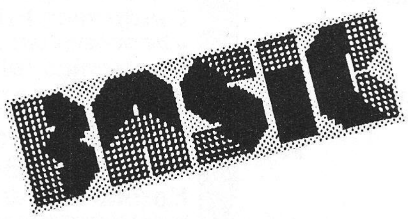
ABS

Type : BASIC functie, rekenkundig

Doel : Het berekenen van de 'absolute' waarde van de uitdrukking die tussen haakjes achter de functie ABS wordt gezet. Simpeler gezegd: ABS maakt een negatieve expressie of dito getal positief. Een dergelijke functie heeft binnen Basic een zekere taak, hoewel je niet meteen kunt zeggen dat het een functie is waar je niet zonder kunt.

Mode : Zowel in programma-mode als direct-mode kan de ABS-functie worden toegepast.

Syntax : ABS (rekenkundige expressie)
De expressie tussen de haken kan natuurlijk niet bestaan uit een string, maar moet uiteindelijk te herleiden zijn tot een numerieke waarde. Een string of foute expressie heeft een foutmelding tot gevolg, zoals 'SYNTAX ERROR', 'TYPE MISMATCH' of 'DIVISION BY ZERO ERROR'. Als de uitkomst van de expressie te groot is, dan ook laat de Commodore het afweten met een 'OVERFLOW ERROR'.
De ABS()-functie kan worden gebruikt aan de rechterkant van een statement waarin waarden aan variabelen worden toegekend (assignments) ofwel na een logische expressie als bijvoorbeeld IF. Ook na een PRINT-opdracht kunnen we de functie ABS() tegenkomen.



Gebruik : Hoewel ABS() een niet erg veel gebruikte functie is, zal ik toch wat voorbeelden van het gebruik bij elkaar proberen te scharrelen, zodat wellicht het nut van een dergelijke functie blijkt. De functie bestaat min of meer bij de gratie van de traditie, want vroeger moest een rekenprocessor voornamelijk worden belast met dit soort activiteiten. Maar hier wat voorbeelden:

I **IF ABS(X) > 100 THEN PRINT "Buiten limiet":STOP**

Bij deze opdracht wordt de waarde die de variabele X kan aannemen, begrensd tussen -100 en 100. Komt deze waarde buiten de gestelde grens, dan zal ABS(X) resulteren in een positieve waarde groter dan +100. Een opdracht als bijvoorbeeld:

Ia **IF ABS(X) > - 10 THEN ......**

heeft natuurlijk geen enkele zin, en zal altijd TRUE als uitkomst hebben.

II **IF ABS (X - X1) > 0.0001 THEN PRINT "AFRONDEN"**

In dit voorbeeld wordt getest of de variabele X in waarde minder afwijkt van de waarde die X1 heeft, dan 0.0001. We kunnen een dergelijke manier van ABS() bijvoorbeeld gebruiken bij het afronden van bedragen en dergelijke. Zoals de fervente programmeur wel zal weten is de Commodore vanwege de wijze van omgaan met de 5 byte grote floating-point rekenkunst in single precision niet al te nauwkeurig. Een opdracht bijvoorbeeld als 7*9 levert bij de Commodore zeker geen 40353607 op, (7*7*7*7*7*7*7*7*7) maar heeft (gek genoeg) 40353607.1 als uitkomst. Om dit type onnauwkeurigheid uit onze rekenprogrammatuur te weren zou de ABS functie kunnen worden gebruikt, omdat we bij dergelijke tests in één opdracht zowel boven als onder de grens van een bepaald bedrag kunnen controleren.

Voorbeeld II kan uiteraard ook als volgt worden geschreven, als we he-

lemaal op de sjieke toer willen schrijven:

IIa **IF ABS (X - X1) > 1E -4 THEN PRINT "AFRONDEN"**

Het aardige bij dit voorbeeld is, dat je ziet dat de grens positief/negatief bij ABS() niet per sé 0 hoeft te zijn, we trekken simpelweg bepaalde bedragen van elkaar af, waardoor het resultaat tot nul evalueert.

III **I%=ABS(100 * SIN(X))**

Het derde voorbeeld van ABS() gebruik zal beter begrepen worden door de wiskundigen onder ons, zeker ook omdat het gebruik van de SINus functie niet al te vaak voorkomt in de huistuin en keukensoftware. Hier wordt gebruik gemaakt van de eigenschap van de SIN() functie, waarbij je moet weten dat de waarde van een SINUS altijd ligt tussen -1 en 1, waardoor in ons voorbeeld de test voor de waarde van I% nooit kan liggen tussen 0 en 10, ongeacht de waarde van de variabele X. Wel zal de waarde van I% in integer-vorm, dus zonder decimalen, al naar gelang de waarde van X een waarde krijgen tussen 0 en 10.

IV **IF ABS(ABS(X-X1) - ABS(Y-Y1)) < 3 THEN GOTO 100: REM OPNIEUW**

Dit voorbeeld maakt een driedubbel gebruik van de functie ABS, eigenlijk alleen om te laten zien, dat dit ook mogelijk is. Het is door mij ooit aangetroffen in een spel-listing, waarbij deze regel is geschreven, om te controleren of twee willekeurig gekozen punten in een matrix (X en Y) niet al te dicht bij elkaar in de buurt lagen. Door beide verschillen weer van elkaar afte trekken kan redelijk worden bepaald of aan het gestelde doel wordt voldaan. Niet bijzonder nuttig, maar leuk om eens te zien. ABS is nu eenmaal de eerste Basic functie die we alfabetisch tegenkomen.

Werkwijze: De ABS() functie in de Basic interpreter behoort tot de allereenvoudigste routines die we kennen. Na het controleren van de geldigheid van de expressie of het getal tussen de haken, wordt de uitkomst in de eerste floating-point accumulator gezet, waarbij de waarde op een ingewikkelde, maar gebruikelijke manier wordt opgedeeld in precies 5 Bytes. Eén van deze bytes, het 'SIGN-byte' heeft een bit gereserveerd voor het 'teken', positief of negatief. Dit bit wordt sowieso op nul gezet door ABS(). En dat is alles. Het getal wat hierdoor uit de floating point accumulator komt is daardoor altijd postief.
De precisie van het getal in de FPA#1 lijdt niet onder de behandeling die door ABS() wordt gegeven. Hele grote getallen worden echter altijd door de accumulator 'gestript' wat de accuratesse enigszins kan aantasten. De waarde wordt dan in veel gevallen geretourneerd in de exponentiële of wis-



kundige notatie.

AND

Type : BASIC operator, binair logisch.

Doel : De 'logische AND' wordt berekend van twee respectieve waarden of expressies. Beide kunnen zowel logisch als rekenkundig zijn, het resultaat van de AND wordt daaraan geconfirmeerd met een rekenkundig resultaat als eerste mogelijkheid. Beide expressies (waarden) moeten wel vooraf worden omgezet naar 2-byte integers. De uitkomst van de AND-operatie is ook een 2-byte integer.
Bij op zich al logische expressies is het resultaat ook logisch, dus 0 of -1, FALSE of TRUE, waardoor van meerdere logische expressies tegelijk de TF-waarde kan worden getest.

Mode : AND kan zowel in programmamode als direct worden gebruikt.

Syntax :
Rekenkundige expressie AND rekenkundige expressie
Logische expressie AND logische expressie
Rekenkundige expressie AND logische expressie (wordt rekenkundig.
De expressie aan beide kanten van AND moeten integer zijn, waardoor de waarde slechts kan liggen tussen -32768 en 32767. Floating point getallen, met decimalen worden automatisch gestript tot integers. Logische expressies vallen automatisch binnen deze grenzen, want die kennen bij de Commodore geen andere waarden dan 0 en -1. Bij gebruik van AND en expressies groter dan de integer limieten wordt een passende foutmelding gegeven. Ook syntax-errors in één van beide expressies worden door de Commodore ondervangen.

Gebruik : Het gebruik van AND zal door iedere programmeur wel eens zijn toegepast. Want met hulp van AND en IF..THEN kunnen heel wat condities tegelijk worden getest. Hoewel voor velen de logische functies van AND misschien wat moeilijker te vatten zijn, ligt het algemene gebruik van AND wel voor de hand:

I **XX = 75 AND 380**
We beginnen met een recht-voor-z'n raap voorbeeld van een AND tussen 2 getallen (integers). Alvast vooruitlopend op de bewerking kan ik zeggen dat de uitkomst in XX uiteindelijk de waarde 72 zal krijgen. Gek misschien, maar waar. Voor allen die het nooit hebben gesnapt, waarom AND met dit soort toch wel vreemde getallen voor de dag komt, zullen we de AND van 75 en 380 wat uiteen rafelen.
Om het nodige van AND te snappen moeten we als eerste de getallen 380 en 75 omzetten naar een binaire notie, net zoals de Commodore dat voor ons doet, door AND erop los te laten. De waarde 380 wordt binair geschreven als %00000001 01111100 terwijl 75 er binair als %00000000 01001011 uitziet. Als tweede zetten we deze binaire getallen onder elkaar:

```
%00000001 01111100   (380)
%00000000 01001011   (75)
```

Nu hoeven we slechts de onderscheiden bits met elkaar te vergelijken aan de hand van de TRUTH-tabel van AND, om te zien wat voor een getal we binair als uitkomst krijgen. Zoals we (hopelijk) weten wordt bij AND de uitkomst 1 (TRUE) als beide bits een waarde 1 hebben. In alle andere gevallen is het resultaat 0.

```
%00000001 01111100   (380)
%00000000 01001011   (75)
-----------------------  AND
%00000000 01001000   (72)
```

II **IF J% < 2 AND C% > 445 THEN PRINT "GELUKT"**

IIa **IF A>0 AND A<100 THEN PRINT "BINNEN DE LIMIET"**

Als tweede voorbeeld een samengestelde AND, waarbij aan beide kanten aan een voorwaarde moet worden voldaan die tot TRUE evalueert, om uiteindelijk de tekst 'GELUKT' of 'BINNEN DE LIMIET' op het scherm te krijgen. Is slechts één van de twee ex-

pressies waar (TRUE) dan treedt ook hier de TRUTH-tabel in werking, waardoor we als het ware krijgen 0 AND 1 -> 0, dus wordt er niet geprint.

III IF X$ = "." AND DEC% = 0 THEN DEC% = LEN(GT$)+1

Een beetje chaotisch lijkt dit voorbeeld misschien wel op het eerste gezicht, maar het verhaal erbij zal wel de nodige duidelijkheid brengen. Deze regel zou kunnen dienen als onderdeel om bij een BASIC-invoerroutine voor getallen de offset te bepalen van een decimale punt binnen het getal GT$. Voordat deze regel wordt uitgevoerd, staat in X$ het te testen karakter uit GT$. DEC% dient als integervlag voor de plaats van een decimale punt.



Zolang geen punt is ingetoetst blijft DEC% de waarde 0 houden, op het moment dat de punt wordt gesignaleerd in X$ krijgt DEC% als waarde de plaats in de string GT$ waar de punt staat. Bovendien is DEC% op dat moment ongelijk 0 en kan dus ook worden gevlagd, dat geen tweede punt mag worden ingevoerd. Dit laatste zit natuurlijk niet in bovenstaand voorbeeld.
Waar het ons hier om gaat is, dat niet alleen getallen kunnen worden ingevoerd, maar ook conditie-tests met string-variabelen en waarden. X$ is duidelijk een string, terwijl we dat helemaal van "." kunnen zeggen. Waar we daarbij wel om moeten denken, is dat de expressie aan iedere kant van AND uiteindelijk moet kunnen evalueren tot een integer-waarde.
Hoewel we met strings werken, zal ook de expressie **X$ = "."** uiteindelijk TRUE (-1) als uitkomst hebben. Geef maar eens **PRINT X$="."** op de computer. De uitkomst is dan geen string, maar gewoon 0 of -1, al naar gelang de inhoud van X$. Bovenstaand voorbeeld is hopelijk wel duidelijk, maar zou heel wat eleganter kunnen worden geschreven als :

IIIa **IF X = 46 AND DEC%=0 THEN DEC% = LEN(GT$) + 1**

Aan deze regel moet vooraf nog de conversie van X=ASC(X$) komen terwijl het getal 46 uiteraard hetzelfde is als ASC("."). De tweede manier komt wat dichter bij de echte taal waarin moet worden geprogrammeerd, maar is voor de minder ervaren Basic-programmeur bijna niet te herleiden tot het doel.

IV **OK = -1 AND JR >1969 AND JR < 2000 AND MD >0 AND MD < 13**

Dit voorbeeld spreekt bijna helemaal voor zichzelf, denk ik. Zeker als erbij wordt gezegd dat de variabele JR staat voor het jaar en MD - hoe kan het anders- voor maand. Het gaat natuurlijk om de vier keren dat AND voorkomt in deze programma-regel. Getest wordt of JR ligt tussen de jaren 1970 en 1999 en de maand tussen 0 en 13, zoals te verwachten is. De enige onbekende is nu nog OK, en daar gaat het uiteindelijk om. In deze regel wordt eigenlijk heel slim gespeeld met AND, omdat je nu in één keer kunt testen of zowel de maand als het jaar goed zijn ingevoerd. Om de uitkomst hiervan ook later nog te kunnen verwerken hadden we iets kunnen bedenken van IF JR AND JR AND MD etcetera THEN OK=-1, maar op die manier moeten we ook de hele IF ... THEN routine erbij halen, en dat maakt de zaak alleen maar langer.

Nu kunnen we volstaan met alleen 4 keer de AND-operater waarbij we beginnen met het op TRUE (-1) zetten van OK. Voldoet de rest van de gestelde condities, dan houdt OK uiteindelijk de waarde -1. Blijkt bijvoorbeeld MD een waarde 15 te hebben dan krijgen we het volgende :

-1 AND -1 AND -1 AND -1 AND 0, wat natuurlijk voor OK 0 oplevert, omdat ergens tussen de AND's een nul als uitkomst van een expressie staat (M<13).

Een dergelijke methode als in dit voorbeeld is erg aan te bevelen, hoewel ik moet toegeven, dat het even wennen is.



Opmerkingen : Bij AND kunnen nog wel eens wat gekke dingen om de hoek komen kijken, vandaar dat het me niet overbodig lijkt nog het een en ander op te merken over deze logische operator.
Als eerste zullen we de TRUTH-tabel van AND nog een keer naar voren halen, zodat daarover geen misverstand kan bestaan:

AND	T	F	1	0
T	T	F	1	0
F	F	F	0	0

Ofwel T AND T = T F AND T = F, en 1 AND 1 = 1 terwijl 0 AND 1 = 0.

De hiërarchie binnen Basic ligt voor AND, OR en NOT helemaal onderin de tabel, dat wil zeggen, dat eerst alle andere operatoren, als **^, * , / , - +** enzovoort worden bewerkt door Basic, waarna pas AND aan de beurt komt. Gelukkig wijkt Basic in deze niet af van talen als C, FORTRAN, COBOL en PASCAL, waardoor het voor de programmeur gemakkelijk wordt gemaakt om vooral complexe expressies foutloos te kunnen uitvoeren. Ook de tests in onze voorbeelden gaan daardoor correct, vooral omdat < en > ook voor AND aan de beurt komen.
Wat de **syntax** betreft is het ook uitkijken met AND, omdat er aan Basic niet al te veel eisen gesteld worden aan deze operatoren. Ze lijken namelijk nogal op variabele-namen, waardoor we in sommige gevallen niet eens een syntax-error krijgen als het fout zit, maar natuurlijk wel een hele serie foute uitkomsten:

100 IF K = X ADN M < 3 THEN QQ = 1000

Deze regel zou een echte 'SYNTAX ERROR' tot gevolg moeten hebben. Dat gebeurt echter niet, want de Commodore trekt zich niet al te veel aan van spaties tussen de verschillende opdrachten en variabelen. De voorgaande programmaregel wordt dan ook geïnterpreteerd als :

100 IFK=XADNM<3THENQQ =1000

ofwel

100 IF K = XA THEN QQ = 1000

De rest van de 'fout' wordt door de interpreter gezien als een aanhangsel van de variabele die begint met X, waarvan de Commodore de eerste twee bytes intern als naam gebruikt. In dit geval dus XA. De uitkomst hangt dus af van de waarde van XA, die 0 is als deze variabele verder in het programma niet voorkomt.

Een leuke Basic-programma regel is de volgende :

IF J < 0 AND J > 200

Hoewel qua syntax volledig correct, zal deze conditie het nooit tot TRUE brengen, vrees ik.
Een andere fout, die erg veel gemaakt wordt bij het gebruik van AND en OR zit in de volgende regel verstopt:

```
100  IF PK AND SS = 0 THEN
     STOP
```

Hier wordt bedoeld te testen of zowel PK als SS de waarde nul hebben. Wat er staat laat echter het geheel al stoppen als alleen SS de waarde 0 heeft. IF PK wordt nu gelezen als IF PK + TRUE en dat is altijd, behalve wanneer PK de waarde nul heeft. Een juiste syntax voor deze regel is natuurlijk:

```
100  IF PK = 0 AND SS = 0 THEN
     STOP
```

Een ernstige AND-(denk)fout, die overigens bijna totaal onvindbaar is, zit in de volgende regel:

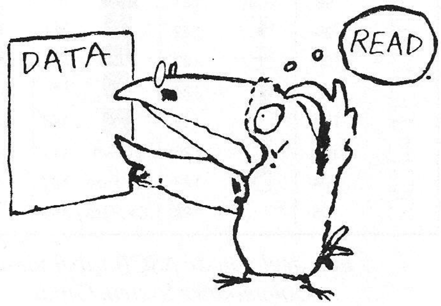
DT = D + DM + 365*J + INT(J/4) + (INT(J/4)* 4=J) AND M > 2

Deze regel komt uit een routine om de juiste weekdag te berekenen. Een paar afleveringen terug hebben we daarvoor een heel programma gepubliceerd. Op die wijze werd ik ook nogal pijnlijk met bovenstaand probleem geconfronteerd, omdat op het eerste gezicht alles nogal normaal lijkt. De bug zit echter in M>2. Want deze expressie na AND is bedoeld om, als de maand groter is dan februari (2), van het totaal 1 dag (bijna automatisch) af te trekken, omdat dan M > evalueert tot -1. Wat echter gebeurt in deze regel is, dat als laatste AND aan de beurt komt. Daardoor wordt alles tussen het =-teken en AND eerst geëvalueerd, waarna dit wordt ge-AND met het resultaat van M>2.
Is de maand januari of februari, dan komt als uitkomst 0 in DT, waardoor de hele boel in de war loopt, zoals te begrijpen is. De oplossing is natuurlijk om nog een paar extra haakjes toe te voegen bij de laatste serie expressies:

DT = D + DM + 365*J + INT(J/4) + ((INT(J/4)* 4=J) AND M > 2)

Nu zal de AND uitsluitend betrekking hebben op het laatste gedeelte, waarin wordt getest of we al dan niet met een schrikkeljaar te maken hebben.

Werkwijze : De manier waarop intern met de AND-operator aan het werk wordt gegaan, speelt zich met name af rond de beide floating-point accumulatoren. Deze 2 keer 6 bytes zijn te vinden in de ZERO-page van de 65XX processor en kunnen daardoor zeer snel worden benaderd en verwerkt. Vandaar dat we met logische ANDs het nodige snel kunnen doen.



De subroutine die door AND wordt gebruikt, is nagenoeg identiek aan die van de OR-operator, alleen wordt bij AND een test-adres vooraf op NUL gezet ($00), terwijl bij OR deze locatie de waarde $FF krijgt toegemeten. Beide operatoren zijn ook nogal aan elkaar verwant. De beide expressies worden vervolgens tot integer-bytes gestript en afzonderlijk verwerkt. Eerst het LOW-byte, daarna het HIGH-byte. De uitkomst van beide wordt opnieuw bewerkt en door de interpreter naar het programma retour gezonden.

ASC

Type : BASIC functie, rekenkundige functie met string-argument

Doel : De Commodore ASCII-waarde wordt berekend van het eerste (meest linkse) karakter van een string-expressie.
Het gebruik van ASCII is onontbeerlijk als afzonderlijke karakters in een string moeten worden getest. Bijvoorbeeld (grafische) karakters voor schermbesturing en opmaak. In het algemeen kan ASC() worden gebruikt, wanneer de numerieke equivalent in een programma gemakkelijker te hanteren is dan de string-equivalent.

Mode : Zowel direct als in programmamode kan de ASC-functie worden toegepast.

Syntax : ASC(string-expressie)

De string kan bestaan uit iedere denkbare geldige string, compleet met de + voor het aan elkaar plakken van strings. De door de Commodore geretourneerde waarden voor ASC liggen tussen 0 en 255. De ASCII-tabel toont de Commodore relaties tussen de verschillende getallen en hun numerieke waarde. Een 'ILLEGAL QUANTITY' error is het gevolg als de string-expressie geen lengte heeft, ofwel als LEN(string)=0.

Gebruik : Als eerste een voorbeeld van een paar regels, waarbij ASC wordt gebruikt bij de afhandeling van een JA/NEE invoerroutine:

I

```
100  GET JN$: IF JN$ = "" THEN
     100
110  JN% = ASC(JN$)
120  IF JN% <> 74 AND JN% <> 78
     THEN PRINT "FOUTE
     INVOER": GOTO 100
130  RETURN : REM JN%=74 of 78
```

Deze routine spreekt bijna voor zichzelf denk ik. De ASC() waarden voor J en N zijn respectievelijk 74 en 78. Voldoet de invoer niet aan één van deze voorwaarden (regel 120) dan wordt opnieuw een invoer gevraagd. Regel 110 bevat de functie ASC, waarbij van de ingevoerde string JN$ eerst de ASCII-waarde naar de numerieke variabele JN% wordt gezet. Let in regel 100 op de test op invoer van een 'lege' string. We moeten hier wel testen op de waarde "", omdat deze (helaas) door Commodore's ASC() functie niet wordt opgevangen. Simpelweg de waarde 0 zou al voldoende zijn. De routine maakt nu, in regel 120, gebruik van de AND operator. We kunnen de routine qua algoritme helemaal veranderen door de OR operator te gebruiken:

Ia

```
100  GET JN$: IF JN$ = "" THEN
     100
110  JN% = ASC(JN$)
120  IF JN% = 74 OR JN% = 78
     THEN RETURN
130  PRINT "FOUTE INVOER"
140  GOTO 100
```

Weinig verschillend in werking, iets korter wellicht en even snel. Maar leuk om deze beide varianten eens naast elkaar te zetten. Natuurlijk hoeft een dergelijke routine niet beperkt te blijven tot invoer van 'J' of 'N'. Op deze manier is het mogelijk om nagenoeg alle toetsen van het toetsenbord, wat invoer betreft, te kunnen testen. Alleen de echte machine-gerichte spe-

ciale toetsen kunnen niet via GET worden ingelezen.

II **FOR J = 1 TO LEN(A$) : POKE 1024+J, ASC(MID$(S$,J,1))-64 : NEXT**

Dit voorbeeld zet de afzonderlijke karakters van een string met hulp van de POKE-opdracht in het computervideo geheugen. Daarvoor moeten de ASC-waarden nog eens met 64 worden verminderd om de scherm-karaktercodes te kunnen emuleren. Hierin is de ASC functie niet te omzeilen, omdat een POKE opdracht uitsluitend numerieke waarden accepteert tussen 0 en 255, precies de range waarin ook de ASCII-karakters zitten, maar dat is uiteraard geen toeval. Kijk ook naar de manier waarop ASC() wordt gebruikt bij het uiteenrafelen van de string S$, met behulp van de MID$() functie. Aan de hand van de waarde van variabele J, wordt telkens één karakter uit de string gehaald en van deze wordt de ASCII-waarde min 64 naar het schermgeheugen gezet.

III **IF ASC(Q$)>192 THEN PRINT CHR$(ASC(Q$)-128)**

In deze programmaregel wordt gekeken of ASC(Q$) een uitkomst heeft die groter is dan 192. Dit betekent dat we te doen hebben met hoofdletters, - althans bij de Commodore, die andere ASCII codes hanteert dan wie ook - deze worden via ASC verminderd met 128, waardoor ze de ASCII-waarde van de juiste kleine letter krijgen. De geconverteerde waarden worden door de CHR$() functie, die precies het omgekeerde doet van de ASC() functie, weer als kleine letters op het scherm gezet.

IV **FOR J = 1 TO LEN(S$) : C(J) = ASC(MID$(S$,J)+J):NEXT**

Deze routine in een FOR....NEXT loop zorgt ervoor, dat de afzonderlijke karakters uit de string S$ worden gecodeerd naar een numerieke waarde die wordt bewaard in de array C(xx). Om het ontcijferen voor onbevoegden 'moeilijker' te maken wordt bij iedere gevonden ASCII-waarde de offset in de string opgeteld. Hoewel een dergelijke code gemakkelijk te breken is, laat het toch een manier zien om met hulp van ASC() hele teksten snel door de computer te laten coderen. Het vraagt echter wat vindingrijkheid, om uit een dergelijke routine een echte onbreekbare code te programmeren, maar het kan. In ieder geval kun je ASC() er prima bij gebruiken.

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
CLR HOME	19	*	42	A	65	X	88
INST DEL	20	+	43	B	66	Y	89
	21	,	44	C	67	Z	90
	22	–	45	D	68	[	91
£	92		115	f4	138		161
]	93		116	f6	139		162
↑	94		117	f8	140		163
←	95		118	SHIFT RETURN	141		164
	96		119	SWITCH TO UPPER CASE	142		165
	97		120		143		166
	98		121	BLK	144		167
	99		122	CRSR	145		168
	100		123	RVS OFF	146		169
	101		124	CLR HOME	147		170
	102		125	INST DEL	148		171
	103		126	Brown	149		172
	104		127	Lt. Red	150		173
	105		128	Dk. Gray	151		174

Een deel van de ASCII tabel uit de Commodore System Guide.

Let in dit voorbeeld ook nog even op de manier waarop MID$() wordt gebruikt, zonder de laatste parameter. In dit soort gevallen wordt door de computer het restant van de string in z'n geheel genomen. Voor de ASC-functie maakt dit niet uit, want deze heeft uitsluitend betrekking op het eerste karakter in de aangeboden stringexpressie, ongeacht de lengte van de string. Voor het computergeheugen maakt een dergelijke methode echter wel uit, omdat op deze wijze het geheugen veel eerder vol raakt en dus meer (onnodig) werk moet worden verzet.

Opmerkingen : De tegenhanger van de ASC() functie is de functie CHR$(). Deze functie doet precies het tegengestelde van de ASC() functie.
Bij de Commodore Basic hebben we te maken met een lastige 'bug' waar het de ASC() functie betreft, die wordt losgelaten op een lege string "". We krijgen dan een foutmelding. Daar moet bij het programmeren terdege rekening mee worden gehouden. Zie hiervoor voorbeeld I. Let er wel op, dat de volgende manier wel een geldige ASC() verwerking is, die ook door de Commodore wordt geaccepteerd:

Ib

```
100 GET JN$: IF JN$ = "" THEN
    JN$=CHR$(0)
110 JN% = ASC(JN$)
120 IF JN% = 74 OR JN% = 78
    THEN RETURN
130 IF JN% <> 0 THEN PRINT
    "FOUTE INVOER"
140 GOTO 100
```

Natuurlijk is het in bovenstaande listing een beetje onzinnig om de verandering van een lege string naar CHR$(0) te forceren, omdat we later alleen maar last hebben van deze waarde (zie regel 130, waar nu een conditie moet worden ingebouwd) maar vanwege eerder gebruik van dit voorbeeld zal het verschil tussen CHR$(0) en de lage sting ("") hiermee wel duidelijk zijn.
De ASCII-waardentabel van Commodore wijkt helaas aanzienlijk af van de ASCII-tabel zoals die wordt gebruikt door alle andere computermerken, zoals bijvoorbeeld de IBM-compatibles en MSX. Bij de 'vreemde' grafische tekens is dat natuurlijk niet zo erg, maar de verschillen tussen de karakters en leestekens, boven- en onderkast, zijn helaas aanzienlijk. Hoewel Commodore de eigen tabel, die dus in feite niets te maken heeft met de echte 'American Standard Code on Information Interchange', al in 1974 heeft ontwikkeld, kan dat toch geen excuus zijn, om nog steeds een afwijkende tabel te gebruiken.

Werkwijze : Als eerste wordt door de Commodore getest of er een legale string-expressie tussen de haken staat van de ASC() functie. Hier wordt dan ook meteen een 'ILLEGAL QUANTITY ERROR' gegenereerd bij de befaamde string met lengte 0. Deze string wordt naar RAM high-memory gezet, als een afzonderlijke variabele, compleet met pointer en lengte-byte op de stack. De primaire ASC() subroutine haalt vervolgens deze bytes weer op van 't stack, en kijkt via de pointer-bytes naar de eerste byte (karakter) van de string. Omdat een string al automatisch in ASC() waarden wordt opgeslagen in RAM, hoeft de ASC()-functie eigenlijk niet anders meer te doen, dan deze waarde toe te kennen aan de gevraagde numerieke variabele, of, via de standaardroutines meteen op het scherm te zetten.
Jammer van de bug, die bijna in alle bestaande Commodore-computertypes voorkomt, maar voor de rest een goed werkende functie.

J.B.

Checksum C-64

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

```
1      rem ******************************
       ***
  2    rem basic loader "SYNTAX.CHECKSUM"
  3    rem na de commando's "run" en "new
       "
  4    rem blijft dit programma in het ge
       -
  5    rem heugen. laad het te testen pro
       -
  6    rem gramma en tik daarna sys 49152
       .
  7    rem ******************************
       ***
10     i=49152  :rem beginadres
20     reada:ifa<0then40:rem data ingelez
       en
30     pokei,a:i=i+1:b=b+a:goto20
40     if b<>16844thenprint"[SHIFT-CLR]fo
       ut[SPACE]in[SPACE]dataregels!":b=0
       :end
50     poke49184,148:poke49185,192
       55 i=49300
60     read a: ifa<0then80
70     pokei,a:b=a+b:i=i+1:goto60
80     if b<>20068thenprint"[SHIFT-CLR]fo
       ut[SPACE]in[SPACE]dataregels![SPAC
       E](vanaf[SPACE]regel[SPACE]240)":b
       =0:end
90     print"data[SPACE]is[SPACE]weggezet
       "
       95 print"checksum[SPACE]testen[SPACE]
       met[SPACE]sys49152"
100    data 165,43,166,44,133,163,134,164
       ,169,147
110    data 32,210,255,160,0,240,3,32,73,
       192
120    data 32,73,192,208,1,96,32,225,255
       ,208
130    data 3,76,116,164,32,81,192,32,73,
       192
140    data 240,12,201,32,240,247,24,101,
       167,133
150    data 167,76,37,192,166,167,169,0,1
       32,168
160    data 32,205,189,169,13,32,210,255,
       164,168
170    data 76,17,192,200,208,2,230,164,1
       77,163
180    data 96,162,0,189,123,192,240,6,32
       ,210
190    data 255,232,208,245,32,73,192,170
       ,32,73
200    data 192,132,168,32,205,189,162,3,
       169,32
210    data 32,210,255,202,208,250,169,0,
       133,167
220    data 164,168,96,82,69,71,69,76,32,
       0
230    data  -1
240    data 165,197,201,3,240,7,201,4,240
250    data 6,76,148,192,76,34,192,169
260    data 147,32,210,255,76,161,192
270    data -1
```

** EINDE LISTING checksum 64 **

REGEL	1	249	REGEL	100	183
REGEL	2	84	REGEL	110	158
REGEL	3	105	REGEL	120	232
REGEL	4	2	REGEL	130	183
REGEL	5	246	REGEL	140	96
REGEL	6	152	REGEL	150	96
REGEL	7	249	REGEL	160	127
REGEL	10	157	REGEL	170	71
REGEL	20	64	REGEL	180	223
REGEL	30	38	REGEL	190	73
REGEL	40	57	REGEL	200	79
REGEL	50	14	REGEL	210	109
REGEL	55	251	REGEL	220	106
REGEL	60	192	REGEL	230	225
REGEL	70	42	REGEL	240	16
REGEL	80	244	REGEL	250	163
REGEL	90	245	REGEL	260	92
REGEL	95	237	REGEL	270	22

PRINT OUT C-64 met o.a. Easybase

Herby C64

Herby een flitsend spel zo noemt de schrijver van het programma het. De schrijver van dit aktiespel is Michiel Geurts uit Horst. In het begin is het nog wel te volgen maar kom je echter een paar levels verder vergeet het maar. Denkt U deze uitdaging aan te kunnen tik dan snel de listing in en probeer het maar. O ja de schrijver stelt zich niet aansprakelijk voor ingegooide beeldbuizen waarbij wij ons dan bij aansluiten.

```
1    poke 56,48:poke55,0:clr
2    naam$="[CTRL-3]michiel":hl=2:a$=ch
     r$(141):dimb$(11):dima$(19):gosub8
     3
3    forr=0to23:readh$:forb=1tolen(h$)s
     tep2:c$=mid$(h$,b,1):d$=mid$(h$,b+
     1,1)
4    gosub5:c=16*g:c$=d$:gosub5:c=c+g:p
     oke49152+t,c:t=t+1:nextb,r:sys4982
     9:goto7
5    g=asc(c$)-48:ifg>9theng=g-7
6    return
7    fori=0to7:poke12288+i,peek(12656+i
     ):next:gosub99:gosub111:gosub 55:g
     osub75
8    print"[SHIFT-CLR]":gosub73:print"[
     HOME][12xCRSR-DOWN][12xCRSR-RIGHT]
     [CTRL-3][CRSR-LEFT]";l
9    poke87,.:poke88,.:poke89,1:poke90,
     1:poke91,1:poke92,23:poke93,32:pok
     e94,.
10   poke49164,232:poke 95,.:poke49701,
     150+l*20:sys49747:ifpeek(2)=1then1
     5
11   ifpeek(89)=peek(91)andpeek(90)=pee
     k(92)then15
12   print"[SHIFT-CLR][COM-6][6xCRSR-DO
     WN]";tab(11);"o.k.[SPACE]!!!":prin
     t"[2xCRSR-DOWN][6xSPACE]je[SPACE]h
     ebt[SPACE]level";l;"overwonnen
13   l=l+1:print"[2xCRSR-DOWN][8xSPACE]
     let[SPACE]op[SPACE]nu[SPACE]komt[S
     PACE]level[COM-7]";l:forop=0to4500
     :next:goto8
14   poke49164,232:poke95,.:poke49701,1
     50+l*10:sys49747:ifpeek(2)=1then15
15   print"[HOME][CRSR-DOWN]";tab(31)"[
     CTRL-3][3xCOM-+][CRSR-DOWN][3xCRSR
     -LEFT][COM-+][CRSR-DOWN][CRSR-LEFT
     ][COM-+][CRSR-DOWN][CRSR-LEFT][3xC
     OM +][2xCRSR-DOWN][3xCRSR-LEFT][3x
     COM-+][CRSR-DOWN][3xCRSR-LEFT][COM
      +][CRSR-DOWN][CRSR-LEFT][COM-+][C
     RSR-DOWN][CRSR-LEFT][CRSR-DOWN][3x
     COM-+][CRSR-DOWN][3xCRSR-LEFT][COM
      +][CRSR-RIGHT][COM-+][CRSR-DOWN][
     3xCRSR-LEFT][3xCOM-+]";
16   print"[CRSR-DOWN][3xCRSR-LEFT][COM
      +][CRSR-RIGHT][COM-+][CRSR-DOWN][
     3xCRSR-LEFT][CRSR-DOWN][3xCOM-+][C
     RSR-DOWN][3xCRSR-LEFT][COM-+][CRSR
     -DOWN][CRSR-LEFT][3xCOM-+][CRSR-DO
     WN][CRSR-LEFT][COM-+][CRSR-DOWN][3
     xCRSR-LEFT][3xCOM-+][CRSR-DOWN][3x
     CRSR-LEFT][CRSR-DOWN][COM-+][CRSR-
     RIGHT][COM-+][CRSR-DOWN][3xCRSR-LE
     FT][3xCOM-+][CRSR-DOWN][3xCRSR-LEF
     T][COM-+][CRSR-RIGHT][COM-+]"
17   fori=0to200:poke53280,i:next:poke5
     3280,.:fori=0to1000:next
18   print"[CTRL-8][SHIFT-CLR][CTRL-7][
     5xCRSR-DOWN][4xCRSR-RIGHT]jammer."
19   print"[2xCRSR-DOWN][4xCRSR-RIGHT]u
     [SPACE]hebt[SPACE]level[SPACE]";l;
     "bereikt":ifl<hlthenforyy=0to5000:
     next:goto30
20   hl=l:print"[2xCRSR-DOWN][4xCRSR-RI
     GHT]u[SPACE]heeft[SPACE]daarmee[SP
     ACE]de[SPACE]hoogste[SPACE]level":
     print"[CRSR-DOWN][4xCRSR-RIGHT]geh
     aald[SPACE]!!!!"
21   print"[CRSR-DOWN][4xCRSR-RIGHT]typ
     [SPACE]uw[SPACE]naam[SPACE]in[SPAC
     E]mbv[SPACE]de[SPACE]joystick.":fo
     ri=1948to1974:pokei,i-1948
22   poke1948,62:next:ii=13:naam$="[CTR
     L 3]":poke1975,31:goto29
23   poke1988+ii,90:if(peek(56320)and4)
     =0thenpoke1988+ii,32:ii=ii-1:ifii<
     0thenii=0
24   if(peek(56320)and8)=0thenpoke1988+
     ii,32:ii=ii+1:ifii>27thenii=27
25   if(peek(56320)and16)>0then23
26   ab$=chr$(64+ii):ifab$="@"then30
27   x=len(naam$):if(ii=27andx>1)orx>10
     thennaam$=left$(naam$,x-1):goto29
28   naam$=naam$+ab$
29   print"[HOME][20xCRSR-DOWN][6xCRSR-
     RIGHT]";naam$;"F[SPACE]":fori=0to1
     00:next:goto23
30   fori=0to2000:gosub99:gosub111:goto
     8
31   dataa658bd12c08d0cc0a659a45aca8659
     845a60e8c8ca88a657bd28c08d22c0a65b
     a45ce886
32   data5b845c60e888cac8a9198c33c018e9
     13a8a90085fba90485fc18a92865fb85fb
     a90065fc
33   data85fc88d0f08a65fb85fba90065fc85
     fc6086fd9818698085fea200a1fd6a60a6
     59a45a20
34   data2cc0a55da20081fb2000c0202cc0a2
     00a1fb855da94481fba45aa6592059c090
     0ba458c8
35   datac004d002a000845860a65ba45c202c
     c0a920a20081fb2016c0202cc0a200a1fb
     c92ed00d
36   data18a905655e855ea900655f855fa945
     81fba45ca65b2059c0900ba457c8c004d0
     02a00084
37   data5760ae00dca55bc90cf012a55cc90c
     f001608a2904f0118a2908f01c608a2901
     f0268a29
38   data02f03160204bc1e001f006e00ff002
     caca4c3dc1204bc1e009f006e017f002e8
     e84c3dc1
39   data204bc1c009f006c017f002c8c84c3d
     c1204bc1c001f006c00ff0028888865b84
     5c202cc0
40   dataa200a94581fb60a65ba45c202cc0a2
     00a92081fba45ca65b608a8665e00c3005
     38a918e5
41   data6560a65b205dc18569a65c205dc185
     6aa659205dc1856ba65a205dc1856ca557
     a6694ab0
42   data02a66aa558a46b4ab002a46c60a65a
     a5586ab002a659e00cf00160206ac1206a
     c1866984
43   data6ae46af0f1a900e46a1002a901856d
```

```
     a658f020e002f029100d2014c2a2e86ab0
     02a2ca4c
44   datafac12014c2a2ca6ab002a2e84cfac1
     2014c2a2b86ab002a2c84cfac12014c2a2
     c86ab002
45   dataa2b88e04c28e05c2a659a45ae8e886
     59845a202cc0a944a20081fb60a659a45a
     202cc0a9
46   data20a20081fba56d60a2a0a000c8d0fd
     e8d0f860a200a55dc945d00bf004c55cd0
     05a20186
47   data0260a55ec93dd0f7a55fc904d0f1a2
     028602602067c02099c12099c020dac020
     2fc2a502
48   datad0242024c2a559c55bd007a55ac55c
     d00160ad00dc2910d00e2099c020dac020
     2fc2a502
49   dataf0046020dac02024c2a559c55bd0bc
     a55ac55cd0b660a6fba4fc4c59c0a9d08d
     bac2a930
50   data8dbdc278a9338501a210a000b900e0
     990030c8d0f7eebac2eebdc2cad0eca937
     850158a9
51   data1f8d18d078a9c38d1503a9008d1403
     5860a910d005a0184cecc2a0008ce2c2a2
     00b90fc3
52   data9d2032c8e8e018d0f44c31eace04c3
     a908d0f6a9088d04c34ce1c2c121121c38
     4484833c
53   data7ee7f7ffff7e3cff81bda5a5bd81ff
     080404699e1010083c76f3ffffff7e3c00
     7e425a5a
54   data 427e00a6fba4fc202cc0a5fda2008
     1fb60
55   be=49822:a$(1)="[SHIFT-CLR][CTRL-6
     ][COM-A][23xSHIFT-*][COM-S]":print
     chr$(142)
56   a$(0)="[6xCRSR-UP]":a$(2)="B......
     ....[3xSPACE].........B"
57   a$(3)="B.[COM-A][8xSHIFT-*][3xSPAC
     E][8xSHIFT-*][COM-S]@B"
58   a$(4)="B.B........[3xSPACE].......
     .B.B"
59   a$(5)="B.B.[COM-A][6xSHIFT-*][3xSP
     ACE][6xSHIFT-*][COM-S].B.B"
60   a$(6)="B.B.B......[3xSPACE]......B
     .B.B"
61   a$(7)="B.B.B.[COM-A][4xSHIFT-*][3x
     SPACE][4xSHIFT-*][COM-S].B.B.B"
62   a$(8)="B.B.B.B....[3xSPACE]....B.B
     .B.B"
63   a$(9)="B.B.B.B.[COM-A][2xSHIFT-*][
     3xSPACE][2xSHIFT-*][COM-S].B.B.B.B
     "
64   a$(10)="B.B.B.B.B..[3xSPACE]..B.B.
     B.B.B"
65   a$(11)="B.B.B.B.B.[COM-A][3xSHIFT-
     *][COM-S].B.B.B.B.B"
66   a$(12)="B[9xSPACE]B[COM-7][3xSPACE
     ][CTRL-6]B[9xSPACE]B"
67   a$(18)="B.B.B.B.B.[COM-Z][3xSHIFT-
     *][COM-X].B.B.B.B.B"
68   a$(13)="B.B.B.B.[COM-Z][2xSHIFT-*]
     [3xSPACE][2xSHIFT-*][COM-X].B.B.B.
     B"
69   a$(14)="B.B.B.[COM-Z][4xSHIFT-*][3
     xSPACE][4xSHIFT-*][COM-X].B.B.B"
70   a$(15)="B.B.[COM-Z][6xSHIFT-*][3xS
     PACE][6xSHIFT-*][COM-X].B.B"
71   a$(16)="B.[COM-Z][8xSHIFT-*][3xSPA
     CE][8xSHIFT-*][COM-X].B"
72   a$(17)="[COM-Z][23xSHIFT-*][COM-X]
     ":return
73   print"[CTRL-6][SHIFT-CLR]":fori=1t
     o12:printa$(i);a$;:next:printa$(12
     );a$;a$(12);
74   printa$;a$(18);:fori=13to17:printa
     $;a$(10-((i-13)*2));a$;a$(i);:next
     :return
75   ce=50688:gosub78:fori=1to23step2:p
     oke251,i:poke252,i:sysbe:gosub77
76   poke252,24-i:sysbe:gosub77:next:fo
     ri=87to95:readd:pokei,d:next:retur
     n
77   poke(peek(254)*256)+peek(253),1:re
     turn
78   fori=ceto50710:reada:pokei,a:next:
     sysce:return:data 169,128,133,252,
     169,0
79   data 133,251,168,145,251,200,208,2
     51,230,252,166,252,224,160,208,243
     ,96
80   data 0,0,1,1,1,23,20,0,0
81   fori=1to10:forii=1tolen(b$(i)):pri
     ntmid$(b$(i),1,ii);:foriii=0to50:
82   nextiii,ii,i
83   c$(1)="[3xCRSR-UP][COM-6][6xCRSR-R
     IGHT][COM-+][SPACE][COM-+][SPACE][
     3xCOM-+][SPACE][3xCOM-+][SPACE][3x
     COM-+][SPACE][COM-+][SPACE][3xCOM-
     +][19xCRSR-RIGHT][COM-+][SPACE][CO
     M +][SPACE][COM-+][3xSPACE][COM-+]
     [SPACE][COM-+][SPACE][COM-+][SPACE
     ][COM-+][SPACE][COM-+][SPACE][COM-
     +]
84   c$(2)="[COM-6][6xCRSR-RIGHT][3xCOM
     +][SPACE][2xCOM-+][2xSPACE][2xCOM
     +][COM--][SPACE][2xCOM-+][COM--][
     SPACE][COM-+][SPACE][2xCOM-+][20xC
     RSR-RIGHT][COM-+][SPACE][COM-+][SP
     ACE][COM-+][3xSPACE][COM-+][SPACE]
     [COM-+][SPACE][COM-+][SPACE][COM-+
     ][SPACE][COM-+][SPACE][COM-+]
85   c$(3)="[COM-6][6xCRSR-RIGHT][COM-+
     ][SPACE][COM-+][SPACE][3xCOM-+][SP
     ACE][COM-+][SPACE][COM-+][SPACE][3
     xCOM-+][SPACE][COM-+][SPACE][3xCOM
     +][18xCRSR-RIGHT][23xCOM-]
86   b$(4)="[CTRL-1][CRSR-DOWN][COM-7][
     2xSPACE]Zorg[SPACE]dat[SPACE][COM-
     6]Herbie[COM-7][SPACE],[SPACE]het[
     SPACE]balletje[SPACE],alle[2xSPACE
     ]":b$(1)=c$(1)
87   b$(5)="[CTRL-1][CRSR-UP][COM-7][2x
     SPACE]puntjes[SPACE]op[SPACE]eet,[
     SPACE]maar......[2xSPACE]pas[SPAC
     E]op,[SPACE]":b$(2)=c$(2)
88   b$(6)="[CTRL-1][CRSR-DOWN][COM-7][
     2xSPACE]zorg[SPACE]dat[SPACE]de[SP
     ACE]computer,de[SPACE]haak,jou[SPA
     CE]niet":b$(3)=c$(3)
89   b$(7)="[CTRL-1][CRSR-DOWN][COM-7][
     2xSPACE]te[SPACE]pakken[SPACE]krij
     gt.[SPACE]Je[SPACE]kunt[SPACE]niet
     [SPACE]meer"
90   b$(8)="[CTRL-1][CRSR-DOWN][COM-7][
     2xSPACE]achteruit[SPACE]als[SPACE]
     je[SPACE]eenmaal[SPACE]in[SPACE]ee
     n[SPACE]gang"
91   b$(9)="[CTRL-1][CRSR-DOWN][COM-7][
     2xSPACE]zit.[SPACE]Duw[SPACE]je[SP
     ACE]op[SPACE]fire,[SPACE]gaat[SPAC
```

```
       E]ie[SPACE]sneller"
92     b$(10)="[CTRL-1][CRSR-DOWN][COM-7]
       [2xSPACE]maar[SPACE]dan[SPACE]kan[
       SPACE]hij[SPACE]maar[SPACE]een[SPA
       CE]baan[SPACE]"
93     b$(11)="[CTRL-1][CRSR-DOWN][COM-7]
       [2xSPACE]springen[2xSPACE](?)[3xSP
       ACE](1[SPACE]min.[SPACE]geduld)[16
       xSPACE][CTRL-1][CRSR-DOWN][COM-6][
       4xSPACE]Michiel[CTRL-1]
94     poke53280,.:poke53281,.:printchr$(
       14);chr$(8);"[SHIFT-CLR][CTRL-1]":
       fori=1to11:
95     print"[CRSR-DOWN][CRSR-LEFT][SPACE
       ][CRSR-UP]":forii=1tolen(b$(i)):pr
       intmid$(b$(i),ii,1);
96     print"[CRSR-DOWN][2xCRSR-LEFT][SPA
       CE][COM-T][CRSR-UP]";:forir=0to1:n
       extir,ii,i:return
99     print"[CTRL-2][SHIFT-CLR]";chr$(14
       2):fori=1028to1059:pokei,70:pokei+
       960,70:nexti:fori=0to24
100    poke1028+i*40,70:poke1059+i*40,70:
       next:print"[HOME][5xCRSR-DOWN]";ta
       b(4)c$(1):
101    printtab(4)c$(2):printtab(4)c$(3):
       print"[2xCRSR-DOWN][10xCRSR-RIGHT]
       [COM-7]joystick[SPACE]in[SPACE]por
       t[SPACE]ii"
102    print"[CRSR-DOWN][14xCRSR-RIGHT][C
       TRL 3]E[COM-7][2xSPACE]herbie":pri
       nt"[CRSR-DOWN][14xCRSR-RIGHT][CTRL
        3]D[COM-7][2xSPACE]haak":kl=11
103    print"[CRSR-DOWN][14xCRSR-RIGHT]hi
       -level[SPACE]:";hl,,,"[CRSR-DOWN][
       3xSPACE]door[SPACE]";naam$
104    print"[HOME][22xCRSR-DOWN]";tab(12
       );"duw[SPACE]op[SPACE]'n[SPACE]toe
       ts."
105    ifkl=11thenkl=12:goto109
106    ifkl=12thenkl=15:goto109
107    ifkl=15thenkl=1:goto109
108    ifkl=1thenkl=11
109    foryi=0to10:gosub110:next:getga$:i
       fga$=""thenpoke646,kl:goto104
110    l=1:return
111    gosub 120:gosub120:print"[4xCOM-+]
       [CRSR-DOWN][4xCRSR-LEFT][COM-+][CR
       SR-DOWN][CRSR-LEFT][4xCOM-+][CRSR-
       DOWN][CRSR-LEFT][COM-+][CRSR-DOWN]
       [4xCRSR-LEFT][4xCOM-+]"
112    gosub120:print"[COM-+][2xSPACE][CO
       M +][CRSR-DOWN][4xCRSR-LEFT][COM-+
       ][2xSPACE][COM-+][CRSR-DOWN][4xCRS
       R-LEFT][4xCOM-+][CRSR-DOWN][CRSR-L
       EFT][COM-+][CRSR-DOWN][CRSR-LEFT][
       COM-+]"
113    gosub120:print"[4xCOM-+][CRSR-DOWN
       ][CRSR-LEFT][COM-+][CRSR-DOWN][3xC
       RSR-LEFT][3xCOM-+][CRSR-DOWN][CRSR
       -LEFT][COM-+][CRSR-DOWN][4xCRSR-LE
       FT][4xCOM-+]"
114    gosub120:print"[COM-][2xCOM-+][CO
       M ][CRSR-DOWN][4xCRSR-LEFT][COM-+
       ][2xSPACE][COM-+][CRSR-DOWN][2xCRS
       R-LEFT][COM-+][COM--][CRSR-DOWN][3
       xCRSR-LEFT][COM-+][COM--][CRSR-DOW
       N][3xCRSR-LEFT][4xCOM-+]"
115    gosub120:print"[COM-][COM-+][CRSR
       -DOWN][CRSR-LEFT][COM-+][CRSR-DOWN
       ][CRSR-LEFT][COM-+][CRSR-DOWN][CRS
       R-LEFT][COM-+][CRSR-DOWN][2xCRSR-L
       EFT][COM-][COM-+][COM-]"
116    gosub120:print"[4xCOM-+][CRSR-DOWN
       ][4xCRSR-LEFT][COM-+][2xSPACE][COM
       +][CRSR-DOWN][4xCRSR-LEFT][COM-+]
       [2xSPACE][COM-+][CRSR-DOWN][4xCRSR
       -LEFT][COM-+][2xSPACE][COM-+][CRSR
       -DOWN][4xCRSR-LEFT][4xCOM-+]"
120    fori=0to500:next:print"[SHIFT-CLR]
       [8xCRSR-DOWN][CTRL-3]";tab(14);:re
       turn
```

** EINDE LISTING herbie **

regel 1	7	regel 40	96	regel 79	60
regel 2	234	regel 41	121	regel 80	253
regel 3	240	regel 42	72	regel 81	153
regel 4	117	regel 43	84	regel 82	144
regel 5	77	regel 44	167	regel 83	110
regel 6	142	regel 45	143	regel 84	159
regel 7	57	regel 46	116	regel 85	60
regel 8	51	regel 47	56	regel 86	112
regel 9	252	regel 48	72	regel 87	64
regel 10	120	regel 49	232	regel 88	6
regel 11	166	regel 50	205	regel 89	21
regel 12	8	regel 51	92	regel 90	183
regel 13	185	regel 52	136	regel 91	246
regel 14	119	regel 53	75	regel 92	253
regel 15	36	regel 54	54	regel 93	172
regel 16	2	regel 55	42	regel 94	197
regel 17	26	regel 56	118	regel 95	72
regel 18	224	regel 57	47	regel 96	246
regel 19	206	regel 58	36	regel 99	235
regel 20	15	regel 59	255	regel 100	199
regel 21	224	regel 60	78	regel 101	20
regel 22	145	regel 61	225	regel 102	224
regel 23	86	regel 62	120	regel 103	76
regel 24	77	regel 63	195	regel 104	16
regel 25	242	regel 64	201	regel 105	230
regel 26	195	regel 65	12	regel 106	234
regel 27	167	regel 66	207	regel 107	184
regel 28	133	regel 67	31	regel 108	87
regel 29	178	regel 68	250	regel 109	6
regel 30	160	regel 69	27	regel 110	247
regel 31	232	regel 70	60	regel 111	11
regel 32	210	regel 71	93	regel 112	25
regel 33	221	regel 72	134	regel 113	110
regel 34	110	regel 73	0	regel 114	91
regel 35	51	regel 74	95	regel 115	194
regel 36	82	regel 75	151	regel 116	95
regel 37	84	regel 76	80	regel 120	9
regel 38	103	regel 77	91		
regel 39	66	regel 78	230		

easybase C64

Nadat U dit programma heeft ingetikt, door de lengte zal dit waarschijnlijk wel de nodige uurtje kosten, beschikt U over een prima database. Het grote voordeel hierbij is dat het programma volledig ikoon gestuurd is. Dit houdt in dat een keuze gemaakt kan worden met de joystick, het onderdeel van uw keuze aanwijzen, op de vuurknop drukken, dat is alles dat nodig is om de informatie op uw scherm te krijgen. Het is een programma van Sander van Schaik uit Bergen.

```
1      rem *** easybase 64 ***
2      rem door  sander-willem
3      rem van schaik,jan 1988
4      rem
10     pl=49384:v=53248:so=53269:sf=11:sp
       =2042:dima$(5,200):n=0:poke53272,2
       3
20     f$(0)="Achternaam":f$(1)="Voornaam
       ..":f$(2)="Adres.....":f$(3)="Woon
       plaats"
30     f$(4)="Postcode..":f$(5)="Telefoon
       .."
40     restore:ir=49152:ip=49416:fort=1to
```

```
6:readsp(t):next:data55,87,119,151,183,215
50 fort=704to767:poket,0:next:forx=49152to49152+287
60 ready$:h=asc(left$(y$,1))-48:l=asc(right$(y$,1))-48
70 y=l+(l>9)*7+16*(h+(h>9)*7):pokex,y:next
80 fort=832to959:reada:poket,a:next:fort=15232to16383:reada:poket,a:next
90 fort=v+4tov+14step2:poket,27:next:sys ir:fort=41to46:pokev+t,1:next
100 pokeso,255:fort=1to6:pokev+3+2*t,sp(t):next
110 gosub860:fort=0to5:pokesp+t,240+t:next
120 syspl,2,4,"Invoeren":syspl,6,4,"Zoeken":syspl,10,4,"Info"
130 syspl,14,4,"Sorteren":syspl,18,4,"Save/Load":syspl,22,4,"Stoppen"
140 gosub 830:onfgoto150,370,400,520,570,790:goto140
150 gosub860:pokesp,252:pokesp+1,253:pokesp+2,254:pokesp+3,246:pokesp+4,sf
160 pokesp+5,sf:syspl,1,4,"Nieuw":syspl,2,4,"record":syspl,5,4,"Record"
170 syspl,6,4,"wijzigen":syspl,9,4,"Record":syspl,10,4,"schrappen"
180 syspl,13,4,"Hoofdmenu"
190 gosub830:onfgoto210,270,310,200:goto190
200 goto 110
210 n=n+1:syspl,2,15,"Record#"n:ifn>200thengoto110
220 syspl,0,15,"INVOEREN":fort=0to5:syspl,t+3,15,f$(t):next
230 fort=0to5:in$="":gosub890:a$(t,n)=in$:next:syspl,10,15,"zo[SPACE]goed[SPACE]?[SPACE](j/n)"
240 getw$:ifw$=""then240
250 ifw$="n"thenn=n-1:goto210
260 goto 110
270 syspl,0,15,"WIJZIGEN":syspl,20,15,"Record?...":t=17:in$=""
280 gosub890:r=val(in$):ifr<1orr>nthengoto110
290 syspl,2,15,"Record#"r
300 fort=0to5:syspl,t+3,15,f$(t):in$="":gosub890:a$(t,r)=in$:next:goto110
310 syspl,0,15,"RECORD[SPACE]SCHRAPPEN":syspl,20,15,"Record?...":t=17:in$=""
320 gosub890:r=val(in$):ifr<1orr>nthengoto110
330 gosub860:pokesp,255:pokesp+1,246:pokesp+2,sf:pokesp+3,sf:syspl,1,4,"Record"
340 syspl,2,4,"weg.":syspl,5,4,"Menu":gosub830:iff=2thengoto110
350 iff<>1then330
360 fort=r+1ton:foru=0to5:a$(u,t-1)=a$(u,t):nextu,t:n=n-1:a$(0,n+1)="":goto 110
370 syspl,0,15,"ZOEKEN":syspl,20,15,"Naam?.....":t=17:in$="":gosub890
380 forj=1ton:fori=0to5:ifa$(i,j)=in$thengosub970
390 nexti,j:goto110
400 gosub860:syspl,0,15,"INFO":pokesp,247:pokesp+1,250:pokesp+2,251
410 syspl,1,4,"Bladeren":syspl,5,4,"Geheugen":syspl,9,4,"Lijst":syspl,13,4,"Menu
420 pokesp+3,246:pokesp+4,sf:pokesp+5,sf:gosub830:onfgoto450,460,490,440
430 goto 420
440 goto 110
450 syspl,0,15,"BLADEREN":forj=1ton:gosub970:next:goto110
460 syspl,0,15,"GEHEUGEN":syspl,3,15,"Er[SPACE]zijn[SPACE]nog[SPACE]";
470 print200-n"records":syspl,4,15,"vrij.":fort=1to250:next
480 wait56320,16,16:goto110
490 syspl,0,15,"LIJST":fort=1tonstep15:foru=0to14
500 syspl,u+3,15,"Rec.#"t+u""a$(0,t+u):nextu:wait56320,16,16:gosub860:nextt
510 goto 110
520 gosub860:syspl,0,15,"SORTEREN":poke53280,0
530 fori=n-1to1step-1:fl=0:forj=1toi:ifa$(0,j)>a$(0,j+1)thenfl=1:gosub560
540 nextj:iffl=0theni=1
550 nexti:poke53280,6:goto110
560 fort=0to5:q$="":q$=a$(t,j):a$(t,j)=a$(t,j+1):a$(t,j+1)=q$:nextt:return
570 pokeso,255
580 sysir:fort=1ton:foru=0to5:ifa$(u,t)=""orleft$(a$(u,t),1)="[SPACE]"thena$(u,t)="?"
590 nextu,t:gosub860:pokesp,248:pokesp+1,249:pokesp+2,238:pokesp+3,239
600 pokesp+4,246:pokesp+5,sf:syspl,9,4,"Save":syspl,13,4,"Load":syspl,2,4,"tape"
610 syspl,6,4,"tape":syspl,10,4,"disk":syspl,14,4,"disk"
620 syspl,1,4,"Save":syspl,5,4,"Load":syspl,17,4,"Hoofdmenu":gosub830:sysip
630 pokeso,0:onfgoto670,740,650,720,640:goto570
640 sysir:goto 110
650 syspl,0,15,"SAVE[HOME][2xSPACE]":fort=1to250:next:wait56320,16,16
660 print"[SHIFT-CLR]":open1,8,2,"@0:data,s,w":goto690
670 syspl,0,15,"SAVE[HOME][2xSPACE]":fort=1to250:next:wait56320,16,16
680 print"[SHIFT-CLR]":open1,1,2,"data"
690 print#1,n:fors=1ton:print#1,a$(0,s):print#1,a$(1,s):print#1,a$(2,s)
700 print#1,a$(3,s):print#1,a$(4,s):print#1,a$(5,s):next:close1
710 print"Wacht[SPACE]even.":goto40
720 syspl,0,15,"LOAD[HOME][2xSPACE]":fort=1to250:next:wait56320,16,16
730 print"[SHIFT-CLR]":open1,8,2,"0:data,s,r":goto760
740 syspl,0,15,"LOAD[HOME][2xSPACE]":fort=1to250:next:wait56320,16,16
750 print"[SHIFT-CLR]":open1,1,0,"data"
760 input#1,n:fors=1ton:input#1,a$(0,s
```

```
     ):input#1,a$(1,s):input#1,a$(2,s)
770  input#1,a$(3,s):input#1,a$(4,s):in
     put#1,a$(5,s):next:close1
780  print"Wacht[SPACE]even.":goto40
790  syspl,0,15,"Wilt[SPACE]u[SPACE]sto
     ppen[SPACE](J/N)"
800  getw$:ifw$=""then800
810  ifw$="j"thenpoke288,0:sys288
820  goto 110
830  pokeso,255:wait56320,16,16:x=peek(
     252)
840  fort=6to1step-1:ifx=>sp(t)thenf=t:
     return
850  next:goto 830
860  print"[SHIFT-CLR][CTRL-2]":poke532
     80,6:poke53281,6:fory=0to24:syspl,
     y,13,"[COM-L][HOME]"
870  syspl,y,0,"[COM-J][HOME]":next
880  syspl,0,0,"[14xCOM-O]":syspl,24,0,
     "[14xCOM-U][HOME]":return
890  syspl,t+3,25,":";
900  getw$:ifw$=""then900
910  w=asc(w$):ifw>95orw=44orw=58orw=59
     then900
920  l=len(in$):iflen(in$)>13thenreturn
930  ifw>31thenin$=in$+w$:printw$;:goto
     900
940  ifw=13andlthenreturn
950  ifw=20andlthenin$=left$(in$,l-1):p
     rintw$;
960  goto 900
970  fort=0to5:syspl,t+3,26,"[14xSPACE]
     ":next
980  fort=0to5:syspl,t+3,15,f$(t)+":"+a
     $(t,j):next:syspl,2,15,"Record#"+s
     tr$(j)
990  wait56320,16,16:return
1000 dataa9,03,8d,15,d0,a9,0d,8d,f8,07,
     a9,0e,8d,f9,07,a9,00,8d,27,d0,a9,0
     8,8d,28
1010 datad0,a9,1a,8d,00,d0,8d,02,d0,85,
     fb,a9,32,8d,01,d0,8d,03,d0,85,fc,a
     9,38,8d
1020 data14,03,a9,c0,8d,15,03,60,a5,fb,
     8d,00,d0,8d,02,d0,a5,fc,8d,01,d0,8
     d,03,d0
1030 dataad,00,dc,c9,7e,f0,21,c9,76,f0,
     27,c9,77,f0,2d,c9,75,f0,32,c9,7d,f
     0,39,c9
1040 data79,f0,3f,c9,7b,f0,45,c9,7a,f0,
     4d,4c,31,ea,00,00,20,c1,c0,4c,31,e
     a,00,00
1050 data00,00,20,c1,c0,20,cb,c0,4c,31,
     ea,00,20,cb,c0,4c,31,ea,00,00,00,2
     0,cb,c0
1060 data20,d5,c0,4c,31,ea,00,00,20,d5,
     c0,4c,31,ea,00,00,00,00,20,d5,c0,2
     0,df,c0
1070 data4c,31,ea,00,20,df,c0,4c,31,ea,
     00,00,00,00,00,00,20,df,c0,20,c1,c
     0,4c,31
1080 dataea,a5,fc,c9,32,f0,02,c6,fc,60,
     00,a5,fb,c9,70,f0,02,e6,fb,60,00,a
     5,fc,c9
1090 datae1,f0,02,e6,fc,60,00,a5,fb,c9,
     12,f0,02,c6,fb,60,20,fd,ae,20,9e,b
     7,8a,48
1100 data20,fd,ae,20,9e,b7,8a,a8,68,aa,
     18,20,f0,ff,20,fd,ae,4c,a0,aa,00,0
     0,00,00
1110 dataa9,31,8d,14,03,a9,ea,8d,15,03,
     60,00,00,00,00,00,00,00,00,00,00,0
     0,00,00
1120 data,,,,,,,,,,,,,,,,,,,,63,192,0
     ,32,128,0,33,,,32,128,,32,64,,40,3
     2,,52
1130 data16,,34,8,,1,16,,,160,,64,,,,,
     ,,,,,,,,,,,,,,,,,,,,,,,,,,,63,192
     ,,63
1140 data128,,63,,,63,128,,63,192,,63,2
     24,,55,240,,35,248,,1,240,,,224,,,
     64
1150 data,,,,,,,,,31,255,248,16,2,8,22
     ,242,8,16,2,24,31,254,24,16,,8,16,
     24,8
1160 data16,36,8,16,36,8,16,24,8,16,,8,
     16,16,8,16,16,8,16,16,8,16,,8,31,2
     55,248
1170 data,16,,,56,,,84,,,16,,,16,,
1180 data31,255,248,16,2,8,22,242,8,16,
     2,24,31,254,24,16,,8,16,24,8
1190 data16,36,8,16,36,8,16,24,8,16,,8,
     16,16,8,16,16,8,16,16,8,16,,8,31,2
     55,248
1200 data,16,,,16,,,84,,,56,,,16,,
1210 data,,,,,,,,2,,,5,,,8,128,,17,64,
     ,10,160,,5,16,,2,72,,1,36
1220 data,,146,,,73,,,36,128,,17,64,,10
     ,64,,4,64,,3,192,,,62,,,,,,,,,,,,
     ,15
1230 data255,128,8,,128,8,,128,8,112,12
     8,8,,128,8,,128,15,255,128,8,,156,
     8,,162
1240 data8,112,162,8,,132,8,,136,15,255
     ,136,8,,136,8,,128,8,112,136,8,,12
     8,8
1250 data,128,15,255,128,,31,255,,16,1,
     ,16,225,,17,241,,19,25,,22,13,,22,
     13,,22
1260 data13,,19,25,,16,49,,16,97,,16,97
     ,,16,97,,16,97,,16,97,,16,1,,16,97
     ,,16
1270 data97,,16,1,,31,255,,,,,,,,,,,,,
     ,7,185,224,4,169,32,7,189,,4,165,3
     2,4
1280 data189,234,,,,1,255,192,2,,192,4,
     249,64,8,2,64,19,228,64,32,8,128,1
     27,241
1290 data,64,18,,71,20,,64,24,,127,240,
     ,,,,0
1300 data7,255,224,4,,32,4,24,32,4,36,9
     6,4,36,32,4,24,32,4,,32,4,16,32,4,
     16,32
1310 data4,16,32,4,,32,7,255,224,,,,7,2
     55,224,4,,32,4,153,32,4,153,32,4,,
     32,4
1320 data255,32,5,,160,7,255,224,0
1330 data,,,,,,,,,,,,,,,,28,,7,255,240,
     4,,16,7,255
1340 data240,4,,16,4,221,144,6,221,176,
     2,221,160,2,73,32,2,73,32,2,73,32,
     3,73
1350 data96,1,73,64,1,73,64,1,,64,1,255
     ,192,,,,,,,,,,,,,,64,,,192,,1,64,
     ,2,127
1360 data192,4,,64,8,,64,4,,64,2,127,19
     2,1,64,,,192,,,64
1370 data,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
     ,,,,,,,,1,248,,1,8,,7,232,,4,40,,3
     1,168
1380 data,16,169,224,126,168,224,66,185
     ,224,90,163,160,66,231,,90,142,,67
```

```
        ,156
1390    data,66,56,,126,48,,,,,,,,,127,255
        ,254,64,,2,64,,2,64,,2,66,60,66,71
        ,36
1400    data226,71,60,226,66,,66,64,,2,71,
        255,226,68,,34,72,,18,127,255,254
1410    data,,,16,,,56,,,84,,,16,,,16,,,1
        6,,,,,,127,255,254,64,,2,64,,2,64,
        ,2,66
1420    data60,66,71,36,226,71,60,226,66,,
        66,64,,2,71,255,226,68,,34,72,,18,
        127
1430    data255,254,,,,,16,,,16,,,16,,,84,
        ,,56,,,16,,,,,,,,,,,,,,,,,,,,,,,2
        55,255
1440    data255,128,,1,128,,17,171,171,169
        ,186,58,137,171,42,145,170,42,145,
        171
1450    data171,129,128,,17,255,255,255,,,
        ,,,,,,,,,,,,,,,,,7,255,,4,1,,5,125
        ,,4,1
1460    data,5,125,,4,1,,5,125,,4,1,,5,125
        ,,4,1,,5,125,,4,1,,5,125,,4,1,,4,1
        ,,4,1
1470    data,7,255,,,,,,,,,,,,,,,,,,,,,,,6
        3,,,33,1,249,45,1,10,33,7,239,237,
        4,42
1480    data33,31,169,33,16,168,63,126,168
        ,,66,184,,90,160,,66,224,,90,128,,
        67,128
1490    data,66,,,126,,,,,,,,,,,,,,,31,128,
        ,16,128,7,254,128,4,34,128,5,186,1
        28,4
1500    data42,128,5,170,128,4,43,128,31,1
        70,,16,238,,126,136,,66,248,,90,12
        8,,66
1510    data128,,90,128,,67,128,,66,,,126,
        ,,,,,,224,,,192,,,160,31,128,16,16
        ,128,7
1520    data254,128,6,34,128,5,122,128,4,1
        70,128,5,106,128,6,43,128,31,170,,
        16,238
1530    data,126,136,,66,248,,90,128,,66,1
        28,,90,128,,67,128,,66,,,126,,,,,,
        0
1540    data,,,,,,128,128,,127,,,99,,,85,
        ,,73,,28,85,28,31,255,252,4,,16,4,
        221
1550    data144,6,221,176,2,221,160,2,73,3
        2,2,73,32,2,73,32,3,73,96,1,73,64,
        1,73
1560    data64,1,,64,1,255,192,0
```

regel 1	66	regel 510	27	regel 1050	208
regel 2	119	regel 520	158	regel 1060	194
regel 3	6	regel 530	10	regel 1070	187
regel 4	143	regel 540	216	regel 1080	122
regel 10	175	regel 550	85	regel 1090	91
regel 20	40	regel 560	113	regel 1100	77
regel 30	122	regel 570	1	regel 1110	36
regel 40	14	regel 580	60	regel 1120	45
regel 50	101	regel 590	72	regel 1130	208
regel 60	179	regel 600	144	regel 1140	4
regel 70	59	regel 610	146	regel 1150	86
regel 80	118	regel 620	43	regel 1160	254
regel 90	250	regel 630	68	regel 1170	35
regel 100	221	regel 640	142	regel 1180	158
regel 110	227	regel 650	181	regel 1190	254
regel 120	189	regel 660	212	regel 1200	35
regel 130	97	regel 670	181	regel 1210	172
regel 140	141	regel 680	191	regel 1220	236
regel 150	192	regel 690	229	regel 1230	251
regel 160	204	regel 700	85	regel 1240	90
regel 170	48	regel 710	87	regel 1250	187
regel 180	191	regel 720	166	regel 1260	145
regel 190	245	regel 730	141	regel 1270	31
regel 200	27	regel 740	166	regel 1280	255
regel 210	197	regel 750	189	regel 1290	16
regel 220	34	regel 760	149	regel 1300	164
regel 230	236	regel 770	25	regel 1310	96
regel 240	143	regel 780	87	regel 1320	75
regel 250	113	regel 790	238	regel 1330	16
regel 260	27	regel 800	145	regel 1340	113
regel 270	158	regel 810	252	regel 1350	157
regel 280	251	regel 820	27	regel 1360	118
regel 290	78	regel 830	227	regel 1370	232
regel 300	193	regel 840	154	regel 1380	169
regel 310	26	regel 850	224	regel 1390	59
regel 320	251	regel 860	162	regel 1400	171
regel 330	137	regel 870	15	regel 1410	99
regel 340	230	regel 880	61	regel 1420	111
regel 350	163	regel 890	15	regel 1430	77
regel 360	4	regel 900	146	regel 1440	117
regel 370	37	regel 910	132	regel 1450	85
regel 380	218	regel 920	171	regel 1460	75
regel 390	150	regel 930	208	regel 1470	21
regel 400	140	regel 940	40	regel 1480	7
regel 410	167	regel 950	182	regel 1490	79
regel 420	246	regel 960	34	regel 1500	195
regel 430	31	regel 970	33	regel 1510	193
regel 440	27	regel 980	123	regel 1520	248
regel 450	39	regel 990	128	regel 1530	220
regel 460	38	regel 1000	69	regel 1540	46
regel 470	162	regel 1010	72	regel 1550	134
regel 480	13	regel 1020	26	regel 1560	8M
regel 490	140	regel 1030	106		
regel 500	185	regel 1040	23		

** EINDE LISTING easybase 64 **

Type kursus 64

Michiel Geurts heeft het al een tijdje in de gaten, om hem heen kunnen vele mensen niet, of niet goed typen. Hij besloot daar dan ook wat aan te doen, en heeft deze uitgebreide type cursus geschreven. Het programma is zo opgebouwd dat elke uitleg eigenlijk overbodig is.

```
1       if peek(49152)=160then44
2       a$="bedankt[SPACE]![4xCRSR-DOWN][7
        xCRSR-LEFT][CTRL-2]voor[SPACE]'t[S
        PACE]run[SPACE]intypen[CRSR-UP][3x
        CRSR-DOWN][6xCRSR-LEFT](met[SPACE]
        twee[SPACE]vingers[SPACE]??)[2xSPA
        CE]"
3       poke204,0:fori=1tolen(a$):wait207,
        1:printmid$(a$,i,1);
4       forii=0tornd(1)*75:nextii,i:poke 2
        07,.:poke204,1:fori=0to2000:next
5       goto 44: rem initialiseren
6       :
7       rem toon voor goed
8       forxx=7to0step-4:pokevv,xx:pokeww,
        129:pokeaa,15:pokehh,40:pokell,200
        :next
9       pokeww,.:pokeaa,.:return
10      :
11      rem toon voor fout
12      pokevv,15:pokeaa,64:pokeaa+1,148:p
        okehh,17:pokell,37:pokeww,33:forxx
        =0to100
13      next:pokeww,32:return
14      fori=1504to2024:pokei,32:next:retu
        rn
15      dd$="":gosub 14:hj$="[HOME][13xCRS
        R-DOWN][5xSPACE][CTRL-2]_[COM-7][S
        PACE]voor[SPACE]menu[SPACE]":print
        hj$
16      print"[CRSR-DOWN][5xSPACE]typ[SPAC
        E]deze[SPACE]zin[SPACE]na:"
17      x=int(rnd(0)*9+1):print"[2xCRSR-DO
```

```
     WN]";tab(2);c$(x)
18   print"[2xCRSR-DOWN]":fg=len(c$(x))
     :ti$="000000"
19   forii=1tofg
20   b$=mid$(c$(x),ii,1):goto22
21   ifd$<>""thengosub12
22   getd$:ifd$<>b$andd$<>"_"then21
23   if d$="_"thengosub14:goto74
24   gosub 7
25   dd$=dd$+d$:print"[HOME][6xCRSR-DOW
     N][2xSPACE][14xCRSR-DOWN]"dd$:next
     ii:d$="":print"[HOME]"
26   op=ti/60:po=int((fg/op)*60):print"
     [HOME][4xSPACE]u[SPACE]typte";po;"
     tekens[SPACE]per[SPACE]minuut.[HOM
     E]"
27   getr$:ifr$=""then27
28   print"[HOME][39xSPACE]"
29   goto 15
30   print"[SHIFT-CLR][CRSR-RIGHT][5xSP
     ACE][CTRL-2][CRSR-RIGHT]typ[SPACE]
     cursus[SPACE]!!!!!!![COM-7]"
31   print"[2xCRSR-DOWN][2xCRSR-RIGHT][
     SPACE]met[SPACE]deze[SPACE]cursus[
     SPACE]kunt[SPACE]u[SPACE]snel"
32   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE]leren[SPACE]typen,[SPACE]mits[
     SPACE]u[SPACE]deze"
33   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE]dingen[SPACE]let:"
34   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE]leg[SPACE]uw[SPACE]vingers[SPA
     CE]van[SPACE]uw[SPACE]linker"
35   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE]hand[SPACE]op[SPACE]de[SPACE]t
     oetsen:"
36   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE][CTRL-2]asdf[COM-7][CRSR-DOWN]
     [4xCRSR-LEFT][4xCOM-T][CRSR-UP][SP
     ACE]en[SPACE]de[SPACE]duim[SPACE]o
     p[SPACE]de[SPACE]spatie-"
37   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE]balk,[SPACE]de[SPACE]vingers[S
     PACE]van[SPACE]uw[SPACE]"
38   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE]rechterhand[SPACE]op[SPACE]de[
     SPACE]toesten:"
39   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE][CTRL-2]:lkj[COM-7][CRSR-DOWN]
     [4xCRSR-LEFT][4xCOM-T][CRSR-UP][SP
     ACE]en[SPACE]de[SPACE]duim[SPACE]o
     p[SPACE]de[SPACE]spatie-"
40   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE]balk,[SPACE]alles[SPACE]moet[S
     PACE]'blind'[SPACE]inge-[2xSPACE]"
41   print"[CRSR-DOWN][2xCRSR-RIGHT][SP
     ACE]typt[SPACE]worden.[3xSPACE](du
     w[SPACE]op[SPACE]toets)[4xCRSR-UP]"
42   geta$:ifa$=""then42
43   run
44   :
45   rem      initalisering
46   :
47   printchr$(142);chr$(8):dim a$(12):
     dimc$(100)
48   vv=54296:ww=54276:aa=54277:hh=5427
     3:ll=54272
49   a$(1)="[COM-A][SHIFT-*][COM-R][SHI
     FT *][COM-R][SHIFT-*][COM-R][SHIFT
      *][COM-R][SHIFT-*][COM-R][SHIFT-*
     ][COM-R][SHIFT-*][COM-R][SHIFT-*][
     COM-R][SHIFT-*][COM-R][SHIFT-*][CO
     M R][SHIFT-*][COM-R][SHIFT-*][COM-
     R][SHIFT-*][COM-S]"
50   a$(2)="B1B2B3B4B5B6B7B8B9B0B+B-BB
     ":rt$(3)="qqwerty":rt$(4)="uuiop@*
     ^"
51   a$(3)="[COM-Z][COM-R][COM-E][COM-R
     ][COM-E][COM-R][COM-E][COM-R][COM-
     E][COM-R][COM-E][COM-R][COM-E][COM
      R][COM-E][COM-R][COM-E][COM-R][CO
     M E][COM-R][COM-E][COM-R][COM-E][C
     OM R][COM-E][COM-R][COM-X]":rt$(1)
     ="aasdfg":rt$(2)="hhjkl:;"
52   a$(4)="[SPACE]BqBwBeBrBtByBuBiBoBp
     B@B*B[SPACE]":rt$(5)="zzxcvb":rt$(
     6)="nnm,./"
53   a$(5)="[SPACE][COM-Z][COM-R][COM-E
     ][COM-R][COM-E][COM-R][COM-E][COM-
     R][COM-E][COM-R][COM-E][COM-R][COM-
     E][COM-R][COM-E][COM-R][COM-E][CO
     M R][COM-E][COM-R][COM-E][COM-R][C
     OM E][COM-R][COM-X][SPACE]":rt$(7)
     ="112345":rt$(8)="667890+-"
54   a$(6)="[2xSPACE]BaBsBdBfBgBhBjBkBl
     B:B;B[2xSPACE]"
55   a$(7)="[2xSPACE][COM-Z][COM-R][COM
      E][COM-R][COM-E][COM-R][COM-E][CO
     M R][COM-E][COM-R][COM-E][COM-R][C
     OM E][COM-R][COM-E][COM-R][COM-E][
     COM-R][COM-E][COM-R][COM-E][COM-R]
     [COM-X][2xSPACE]"
56   a$(8)="[3xSPACE]BzBxBcBvBbBnBmB,B.
     B/B[3xSPACE]"
57   a$(9)="[3xSPACE][COM-Z][SHIFT-*][C
     OM E][SHIFT-*][COM-E][SHIFT-*][COM
      E][SHIFT-*][COM-E][SHIFT-*][COM-E
     ][SHIFT-*][COM-E][SHIFT-*][COM-E][
     SHIFT-*][COM-E][SHIFT-*][COM-E][SH
     IFT *][COM-X][3xSPACE]"
58   spatie$="[39xSPACE]"
59   poke53280,.:poke53281,.:qq=0:gosub
     60:goto66
60   print"[COM-7][SHIFT-CLR][CRSR-DOWN
     ]"
61   fori=1to09:printtab(6);a$(i):next
62   fori=55423to55450step2:pokei,qq:ne
     xt
63   fori=55500to55531step2:pokei,qq:ne
     xt
64   fori=55581to55610step2:pokei,qq:ne
     xt
65   fori=55660to55691step2:pokei,qq:ne
     xt:return
66   fori=49152to49244:reada:pokei,a:ne
     xt:poke49165,2:sys 49152
67   fori=1to11:readc$(i):next
68   fori=0to60:poke49165,i:sys49152:sy
     s49228:fora=0to20:next:next
69   data 160,2,169,4,141,11,192,162,25
     4,189,0,5,201,2,240,10,202,208,246
70   data 238,11,192,136,208,238,96,24,
     138,109,10,192,133,251,24,173,11
71   data 192,105,212,133,252,120,169,5
     4,141,20,3,169,192,141,21,3,88,96
72   data 165,2,201,10,208,11,169,0,133
     ,2,173,18,208,160,0,145,251,230
73   data 2,76,49,234,169,1,160,0,145,2
     51,169,234,141,21,3,169,49,141,20,
     3,96
74   print"[HOME][13xCRSR-DOWN][4xSPACE
     ][CTRL-2][13xSPACE]menu[COM-7]"
```

```
75   print"[2xCRSR-DOWN][10xSPACE]1)[SP
     ACE]gewoon[SPACE]intypen."
76   print"[CRSR-DOWN][10xSPACE]2)[SPAC
     E]instrukties."
77   print"[CRSR-DOWN][10xSPACE]3)[SPAC
     E]stoppen."
78   geta$:a=val(a$):onagoto82,30,80
79   goto 78
80   print"[SHIFT-CLR][2xCRSR-DOWN]tot[
     SPACE]ziens[SPACE]!!":print"klaar.
     ":poke768,123:poke769,164:end
81   :
82   rem cursus
83   print"[SHIFT-CLR][2xSPACE]met[SPAC
     E]deze[SPACE]cursussen[SPACE]kunt[
     SPACE]u[SPACE]systematies"
84   print"[CRSR-DOWN][2xSPACE]leren[SP
     ACE]typen,[SPACE]gaat[SPACE]u[SPAC
     E]pas[SPACE]door[SPACE]met[SPACE]d
     e"
85   print"[CRSR-DOWN][2xSPACE]volgende
     [SPACE]cursus[SPACE]als[SPACE]u[SP
     ACE]er[SPACE]een[SPACE]helemaal"
86   print"[CRSR-DOWN][2xSPACE]beheerst
     .":print"[2xCRSR-DOWN][5xSPACE]cur
     sus[SPACE]0)[SPACE]asdfg[4xSPACE]"
87   print"[5xSPACE]cursus[SPACE]1)[SPA
     CE]hjkl:;[24xSPACE]cursus[SPACE]2)
     [SPACE]qwerty"
88   print"[5xSPACE]cursus[SPACE]3)[SPA
     CE]uioop@*[23xSPACE]cursus[SPACE]4
     )[SPACE]zxcvb"
89   print"[5xSPACE]cursus[SPACE]5)[SPA
     CE]nm,./[25xSPACE]cursus[SPACE]6)[
     SPACE]123456"
90   print"[5xSPACE]cursus[SPACE]7)[SPA
     CE]67890+-[23xSPACE]cursus[SPACE]
     8)[SPACE]zinnen"
91   print"[12xSPACE]9)[SPACE]terug[25x
     SPACE][2xCRSR-DOWN][CTRL-2][SPACE]
     wat[SPACE]kiest[SPACE]u[SPACE]?[SP
     ACE](0-9)
92   getcu$:cu$=cu$+"a":cu=asc(cu$)-47:
     ifcu<1orcu>10then92
93   ifcu=9thenprint"[SHIFT-CLR]":qq=1:
     gosub60:goto15
94   ifcu=10thenrun
95   rr$=rt$(cu):qq=1:gosub60:print"[HO
     ME][16xCRSR-DOWN][12xCRSR-RIGHT][C
     TRL 2]_[COM-7][SPACE]voor[SPACE]me
     nu"
96   ee=len(rr$):er=int(rnd(1)*ee+1):et
     $=mid$(rr$,er,1):ifet$=te$then96
97   te$=et$:print"[HOME][13xCRSR-DOWN]
     [12xSPACE]typt[SPACE]u[SPACE]de[SP
     ACE][CTRL-2]";et$;"[SPACE][COM-7].
     ..":gosub102
98   getb$:ifb$=et$thengosub8:gosub102:
     goto96
99   if b$="_"thengoto 82
100  ifb$<>""thengosub12
101  goto98
102  d=peek(646):kl=49152
103  print"[HOME][CTRL-1]";et$:b=peek(1
     024):poke49165,peek(1024):sys49228
     :syskl:print"[HOME][SPACE]"
104  poke646,d:return
200  data"computers[SPACE]zijn[SPACE]on
     tzettend[SPACE]handig"
210  data"die[SPACE]bananen[SPACE]koste
     n[SPACE]fl[SPACE]12,50."
220  data"ik[SPACE]zeg[SPACE]altijd[SPA
     CE]maar[SPACE]:[SPACE]tjekaa."
230  data"1[SPACE]is[SPACE]ongeveer[SP
     ACE]fl[SPACE]4,50."
240  data"12*45=540,[SPACE]2*12*3=840,[
     SPACE]34+8=42"
250  data"dit[SPACE]is[SPACE]gemaakt[SP
     ACE]door[SPACE]michiel[SPACE]geurt
     s."
260  data"@[SPACE]dit[SPACE]teken[SPACE
     ]kom[SPACE]je[SPACE]nooit[SPACE]te
     gen."
270  data"weet[SPACE]jij[SPACE]waar[SPA
     CE]willem[SPACE]wever[SPACE]woont"
280  data"lees[SPACE]jij[SPACE]ook[SPAC
     E]commodore-info."
290  data"ik[SPACE]typ[SPACE]al[SPACE]h
     eel[SPACE]snel."
300  data"eind."
500  rem      u kunt deze data regels
     .
510  rem      gewoon aanvullen.
     .
520  rem      u moet dan wel regel 67
     .
530  rem      aanpassen   (fori=1to...)
590  :
```

** EINDE LISTING type cursursus **

regel 1	251	regel 25	86	regel 49	83	regel 73	110	regel 97	92
regel 2	96	regel 26	16	regel 50	207	regel 74	161	regel 98	153
regel 3	175	regel 27	88	regel 51	36	regel 75	125	regel 99	224
regel 4	65	regel 28	240	regel 52	167	regel 76	220	regel 100	48
regel 5	138	regel 29	239	regel 53	107	regel 77	161	regel 101	250
regel 6	58	regel 30	18	regel 54	253	regel 78	208	regel 102	49
regel 7	52	regel 31	245	regel 55	221	regel 79	248	regel 103	218
regel 8	207	regel 32	20	regel 56	235	regel 80	52	regel 104	111
regel 9	20	regel 33	252	regel 57	8	regel 81	58	regel 200	126
regel 10	58	regel 34	32	regel 58	224	regel 82	116	regel 210	20
regel 11	83	regel 35	199	regel 59	59	regel 83	59	regel 220	50
regel 12	60	regel 36	237	regel 60	27	regel 84	25	regel 230	208
regel 13	90	regel 37	166	regel 61	133	regel 85	4	regel 240	77
regel 14	225	regel 38	212	regel 62	165	regel 86	74	regel 250	117
regel 15	166	regel 39	234	regel 63	161	regel 87	230	regel 260	255
regel 16	205	regel 40	27	regel 64	168	regel 88	227	regel 270	238
regel 17	201	regel 41	41	regel 65	119	regel 89	189	regel 280	226
regel 18	173	regel 42	51	regel 66	134	regel 90	252	regel 290	99
regel 19	39	regel 43	138	regel 67	49	regel 91	178	regel 300	21
regel 20	133	regel 44	58	regel 68	31	regel 92	10	regel 500	88
regel 21	50	regel 45	97	regel 69	141	regel 93	58	regel 510	96
regel 22	40	regel 46	58	regel 70	73	regel 94	103	regel 520	222
regel 23	15	regel 47	181	regel 71	116	regel 95	40	regel 530	69
regel 24	196	regel 48	138	regel 72	242	regel 96	162	regel 590	5

PRINT OUT met o.a.Duikboot

Checksum c16

```
10   rem *******************************
     ***
20   rem syntax.checksum
30   rem voor c-16 & plus/4
40   rem
50   rem syntax testen met 'sys 1536'
60   rem
70   rem v.851128.16        jan bodzinga
80   rem *******************************
     ***
90   i=1536       :rem beginadres
100  reada:ifa>=0then pokei,a:i=i+1:got
     o100
110  print"data[SPACE]is[SPACE]weggezet
     "
120  print"cheksum[SPACE]printen[SPACE]
     met[SPACE]'sys[SPACE]1536'
130  end
200  data 165, 43,166, 44,133
210  data  31,134, 32,169,147
220  data  32,210,255,160,  0
230  data 240,  3, 32, 73,  6
240  data  32, 73,  6,208,  1
250  data  96, 72,152, 32,131
260  data   6,168,104,234, 32
270  data  81,  6, 32, 73,  6
280  data 240, 12,201, 32,240
290  data 247, 24,101,252,133
300  data 252, 76, 37,  6,166
310  data 252,169,  0,132,253
320  data  32, 95,164,169, 13
330  data  32,210,255,164,253
340  data  76, 17,  6,200,208
350  data   2,230, 32,177, 31
360  data  96,162,  0,189,123
370  data   6,240,  6, 32,210
380  data 255,232,208,245, 32
390  data  73,  6,170, 32, 73
400  data   6,132,253, 32, 95
410  data 164,162,  3,169, 32
420  data  32,210,255,202,208
430  data 250,169,  0,133,252
440  data 164,253, 96, 82, 69
450  data  71, 69, 76, 32,  0
460  data   0, 72,138, 72, 32
470  data 225,255,240,251,104
480  data 170,104, 96, -1
```

** EINDE LISTING checks16

REGEL 10	249	REGEL 300	118
REGEL 20	247	REGEL 310	204
REGEL 30	121	REGEL 320	165
REGEL 40	143	REGEL 330	252
REGEL 50	75	REGEL 340	106
REGEL 60	143	REGEL 350	98
REGEL 70	8	REGEL 360	163
REGEL 80	249	REGEL 370	45
REGEL 90	103	REGEL 380	0
REGEL 100	2	REGEL 390	58
REGEL 110	245	REGEL 400	108
REGEL 120	237	REGEL 410	159
REGEL 130	128	REGEL 420	245
REGEL 200	210	REGEL 430	202
REGEL 210	208	REGEL 440	176
REGEL 220	142	REGEL 450	12
REGEL 230	1	REGEL 460	54
REGEL 240	3	REGEL 470	43
REGEL 250	157	REGEL 480	1
REGEL 260	155		
REGEL 270	215		
REGEL 280	186		
REGEL 290	248		

Zon en maan C16

Hebben we de vorige keer de amiga een beetje in het zonnetje gezet door een vrij uitgebreid programma met uitvoerige uitleg , ditmaal wilden we de c 16 in het zonnetje zetten en wel met het programma Zon en Maan van Hans Quene'. Hij heeft een programma geschreven voor de c 16 want 16 kilobyte is ruim voldoende om te weten tot hoe laat het op de plaats van bestemming licht zal zijn. De dag te bepalen dat de zon weer je raam binnen schijnt, na te gaan of er volle maan was op d-day en tot op de minuten nauwkeurig voorspellen wanneer de vogels gaan fluiten, de lantaarnpalen uit zal gaan en de teunisbloem zich opent. Natuurlijk kan je gewoon wat data intikken en zo een heleboel te weten komen over de hemelmechanica. De resultaten van dit programma heeft de auteur langdurig vergeleken met tabellen van erkende instellingen en voor de op- en ondergangstijden vond hij geen grotere afwijking dan 1 minuut, voor de azimuth vond hij een maximaal verschil van een halve graad. de afstand tot de zon is correct als hij wordt afgerond tot op vier cijfers achter de komma. Het zelfde geldt voor de eerste twee cijfers van het verlicht gedeelte van de maan. De hoogten, schijnbare straal en afstand maan heeft hij niet systematisch kunnen onderzoeken. toch moeten zij redelijk kloppen omdat anders de op- en ondergangstijden foutief zouden zijn. Voor de goede orde vermeld de schrijver dat de formules heeft uit de Grunlagen. De schrijver heeft een uitgebreide handleiding geschreven die wij dan ook letterlijk overnemen. Inputgegevens:
Het jaartal moet volledig worden opgegeven dus 1988 inplaats van '88 jaartallen van voor 1583 worden niet uitgerekend. de maand moet in cijfers worden opgegeven.

Wereld tijd:

Het programma maakt uitsluitend gebruik van wereldtijd (UT) die eerst in uren (van 0 tot 32) en dan in minuten moet worden opgegeven. Op enkele seconden na is de wereldtijd hetzelfde als Greenwich Mean time (GMT). Onze tijd loopt 1 uur voor op de GMT, de zomertijd 2 uur. Dit moet dus van de tijd die onze klok aangeeft worden afgetrokken om wereldtijd te verkrijgen. Voor de lokale tijd op een ander punt op aarde raadpleegt men een tijd zonekaart. In het algemeen geldt dat voor elke 15 graden die men oostwaarts reist de klok een uur vooruit, en voor elke 15 graden westwaards een uur achteruit moet worden gezet. Maar hier bestaan vele uitzonderingen op.

Geografische lengte en breedte:

Het systeem van geografische lengte en breedte legt elke plaats op de aarde met behulp van twee koördinaten eenduidig vast. Lengte en breedte worden daarbij in graden gemeten. De lengte loopt van +90

(noordpool) via 0 (evenaar) naar -90 (zuidpool). De breedte loopt van -180 tot +180, waarbij de nulmeridiaan voor Greenwich (vlakbij Londen) loopt. Vermoedelijk ligt uw woonplaats niet ver van een van deze plaatsen

Amsterdam	**52,38 -4.83**
Antwerpen	**51.23 -4.38**
Arnhem	**51.98 -5.92**
Brussel	**50.85 -4.35**
Eindhoven	**52.43 -5.50**
Gent	**51.08 -3.73**
Groningen	**53.22 -6.57**
Den Haag	**52.08 -4.30**
Leeuwarden	**63.20 -5.68**
Maastricht	**50.85 -5.68**
Utrecht	**52.08 -5.13**
Vlissingen	**51.43 -3.53**
Zwolle	**52.30 -6.08**

Ook mag een tabel van enkele wereldsteden niet ontbreken :

Londen	**51.50 .08 (0)**
Moskou	**55.75 -37.60 (-3)**
Paramaribo	**5.83 55.17 (-3 1/2)**
Parijs	**48.87 -2.33 (+1)**
New York	**40.75 74.00 (-5)**
Tokio	**35.43 -139.75 (+9)**
San Fransisco	**37.25 122.50 (-8)**

Tussen haakjes staat het aantal uren dat men bij de wereldtijd moet optellen of aftrekken om de lokale wintertijd te verkrijgen. Wie zelf in een atlas lengte en breedte op wil zoeken moet weten dat het daar gebruikte systeem op twee punten af wijkt Ten eerste spreekt de atlas van noorden-en zuiderbreedte, en ooster- en westerlengte.
Het programma onderscheidt de halfronden door middel van positieve en negatieve getallen. Zuiderbreedten en oosterlengten moeten als negatieve getallen worden ingevoerd. Ten tweede maakt de atlas gebruik van minuten. Er gaan zestig minuten in een graad. Deze moeten naar een tiendelige breuk worden omgerekend. Om kort te gaan: 38.53' ZB wordt -38.88. Veel atlassen hebben ook een index waarin lengten en breedten vermeld staan, zodat die dan niet van een kaart hoeven te worden afgeleid. Tot slot moet nog worden opgemerkt dat het programma vastloopt als een breedte van plus of min 90 wordt ingevoerd. Dat gebeurt niet bij afstanden op enkele kilometers van de pool, b.v. 89.99.

Outputgegevens:

Azimuth en hoogte. Met behulp van azimuth en hoogte kan je vastleggen waar "op de hemelkoepel" een hemellichaam zich bevindt, bevond of zal bevinden. Azimuth en hoogte worden in graden en minuten gemeten. De azimuth geeft de kompasrichting aan waarin men moet kijken om het hemellichaam waar te nemen. Als de azimuth 0 tot 360 graden is, dan staat het hemellichaam in het noorden, bij 90 graden in het oosten, bij 180 graden in het zuiden en bij 270 graden in het westen. Als een hemellichaam een hoogte heeft van 90 graden dan staat deze recht boven de waarnemer (in het zenith), bij een hoogte van 0 graden op de horizon, terwijl bij negatieve waarden het hemellichaam onder de horizon staat en dan dus niet waar te nemen is. Dit laatste is bij de zon toch een practisch gegeven omdat het aantal graden dat de zon onder de horizon staat een objectieve maat voor de schemertoestand is. Ik zal me beperken tot een zonsondergang, maar in omgekeerde volgorde geldt alles ook voor een zonsopgang. -0.83 ofwel 50 minuten. Definitie zonsondergang. -6.00 eind burgelijke schemering. -12.00 eind nautische schemering -18.00 eind astronomische schemering Tot het eind van de burgelijke schemering is het nog vrij licht. Buiten kan men nog wel lezen, binnen lukt dat niet meer. Men doet de lampen aan en de helderste sterren worden zichtbaar. Tussen het eind van de burgelijke schemering en het eind van de nautische schemering wordt lezen onmogelijk en verliezen we het vermogen om kleuren te onderscheiden. Tot het eind van de astronomische schemering kan wie een vrij uitzicht heeft nog zien waar de zon onderging, al wordt dit steeds moeilijker Na het eind van de astronomische schemering zijn zelfs de zwakste sterren in principe zichtbaar. In Nederland wordt dit van half mei tot half juli niet bereikt. Uit de azimuth en hoogte van de zon kunnen ook gegevens over de schaduw afgeleid worden. Een paal a meter, loodrecht op het aardoppervlak, werpt een schaduw van a/tan(hoogte) meter. Verder is het niet mogelijk in te zien dat de richting van die schaduw precies tegengesteld is aan die van de azimuth.

Op komst en ondergang:

De opkomst- en ondergangstijden van zon en maan worden opgegeven in uren en minuten, en in wereldtijd. De tijden worden niet berekend maar benaderd. Daarom duurt het uitrekenen van de opkomst en ondergangstijd van de maan ca 25 seconden. De computer probeert een zodanige tijd te vinden dat de zon 50 minuten onder de horizon staat. Op deze manier heeft de zonsondergang betrekking op de bovenste rand van de zon, en niet op het middelpunt. Bovendien verbuigt de lucht in de dampkring het zonlicht, zodat het lijkt alsof de zon wat hoger staat dan in werkelijkheid het geval is. Dit effect treedt het sterkst op als het licht een lange weg moet afleggen, dus als de zon laag staat. Boven de poolcirkel kan het voorkomen dat de zon niet opkomt en niet ondergaat. Dit wordt aangegeven met het woordje "niet". De hoogte van de zon geeft dan uitsluitsel of de zon er die dag nu juist wel of niet juist is. Ook de maan kent een soort "poolcirkel", maar deze verplaatst zich veel meer. In voorkomende gevallen verschijnt ook hier het woordje "niet" De maan trekt zich niets aan wat wij als dag en nacht beschouwen. Ze komt ongeveer om de 25 uur op en gaat ongeveer om de 25 uur onder. Daarom is er elke maand wel een etmaal waarop de maan niet opkomt en een ander etmaal waarop ze niet ondergaat. Dit wordt dan aangegeven met het woordje "niet", wat gelezen kan worden als "pas morgenochtend vroeg". Dezelfde "eigenzinnigheid" is er de oorzaak van dat op de helft van de dagen de maan eerder ondergaat dan opkomt. In die tussenliggende tijd is er dus geen maan.

Afstanden en schijnbare straal:

De afstanden van de aarde tot de zon wordt opgegeven in Astronomische Eenheden. Een A E is gelijk aan 149597870 km. Dit kan men beschouwen als de gemiddelde afstand van de aarde tot de zon. Met behulp van het gegeven dat de lichtsnelheid 299792458 meter per seconde bedraagt kan men zelf berekenen dat het zonlicht er ruim acht minuten over doet om ons te bereiken. De afstand aarde-maan wordt opgegeven in kilometer. Het is misschien aardig om te weten dat sinds astronauten een spiegeltje op de maan hebben geplaatst, men met behulp van laserlicht de afstand tot op 8 cm nauwkeurig kan meten. Omdat die afstand nogal varieert verandert ook de grootte van schijfje die wij waarnemen. Bij sommige zonsverduisteringen bedekt de maan de zon geheel, bij andere blijft een ring van licht zichtbaar. Het programma geeft de schijnbare straal in graden.

Verlicht gedeelte:

Hoewel in werkelijkheid de zon steeds vijftig procent van het maanoppervlak verlicht, lijkt het voor de aardse, waarnemer alsof de maan een schijfje is, waarvan een kleiner of groter gedeelte verlicht kan zijn. Bij volle maan is het verlicht gedeelte 1.00 bij .50 is de maan "halfvol" en bij 0.0 is het nieuwe maan. Als gevolg van onnauwkeurigheid kan het wel eens voorkomen dat de computer een verlicht gedeelte opgeeft dat iets groter is dan 1. Wie wil weten of het linker of rechter gedeelte verlicht is vraagt de gegevens van dezelfde plaatst, maar een paar uur later. Is het gedeelte toegenomen dan was het rechter gedeelte verlicht, anders het linker. In zuidelijke streken lijkt de maansikkel onderuit te zakken; op de evenaar vormt de maansikkel een boortje of een boogje.

Aanvullende opmerkingen:

Controleren van de listing. allereerst moet de checksum in orde zijn. Maar zoals bekend signaleert een checksum programma geen fout als twee karakters onderling verwisseld worden. dat laatste kan bij dit programma nogal gemakkelijk gebeuren omdat er veel getallen in voorkomen. Voer daarom deze programma regels in:

```
195 print"t=";t:print"ts=";ts:print"l=";l:
    print"rz=";rz:print "b=";b:goto 210
265 print"co=";co:print"l=";l:print"b=";b:
    print"f=";f:goto280
```

De checksums zijn 81 en 43. Run dan met deze gegevens jaar 1963, maand 7, dag 10, wereldtijd 7, minuten 30, geo breedte 52.15, geo lengte -4.5. Als de waarden in onderstaande tabel niet precies overeenstemmen met die op het schermpje dan zit er vermoedelijk een fout in de gosubs die beginnen op de achter de variabele vermelde regelnummers.

var	waarde	gosubs
t	.635203628	400,470
ts	.776851879	470
l	1.87351279	630
rz	1.01665821	710
b	6.28318283	760
co	-17576.319	1260, 1320
l	5.7361403	1260, 1320
b	6.22598387	1260, 1320, 1400
f	.78837948	1470

Werk de variabele in de gegeven volgorde af, omdat als bijvoorbeeld de variabele t fout is, alle andere variabele gegarandeerd ook een foute waarde krijgen. Als alles klopt kunnen de regels 195 en 265 weer verwijderd worden.

Persoonlijke aanvulling op het programma:

Als je het leuk vindt als het programma ook meldt hoeveel dagen je oud bent en hoeveel kilometer de opgegeven 'plaats hemelsbreed van je woonplaats is verwijderd kan je als volgt te werk gaan:

1) voer deze programma regels in:

```
145 print int (jd) :print
    sin(gb);cos(gb);-le:stop
```

(checksum is 173.)

2) Run het programma met je geboorte datum en je woonplaats, en een willekeurige wereldtijd. Schrijf de vier getallen over en noem ze a,b1,b2 en b3. 3) Voer nu deze regels in:

```
593 print"leeftijd";int(jd)-A;"dagen";
597 di=b1*sin(gb)[b2@cos(fnr(le+b3))
598
    di=int((fnc(di)*6378.140)+.5):print"afsta
    nd";di;"kilometer"
```

Gebruik daarbij in regel 593 en 597 niet daadwerkelijk de variabelen a,b1,b2 en b3 maar vervang deze door de onder 2 gevonden getallen.

Ascendant:

Met zekere tegenzin geef ik voor het genoegen van hen die geen kans zien om hun betrekkingswaan door middel van zelfreflectie te verdrijven een tabel waarmee de astrologische ascendant bepaald kan worden. De tabel geldt uitsluitend voor Nederland en Belgie. eerst moet deze programma regel ingevoerd worden:

```
145 print int(100*ts+.5)/100{stop
```

(Checksum is 152)

Run vervolgens het programma met plaats, datum en wereldtijd van geboorte. Op het scherm verschijnt nu een getal met twee cijfers achter de komma. Is dit getal bijvoorbeeld 1.98 dan is de ascendant weegschaal omdat dit getal ergens tussen de waarden 1.57 en 2.31 in ligt. Zou dit getal precies 1.57 zijn dan is de ascendant om diverse redenen onbekend en kan zowel maagd als weegschaal zijn. Kreeft 0.08 leeuw 0.82 maagd 1.57 weegschaal 2.31 schorpioen 3.05 boogschutter 3.72 steenbok 4.18 waterman 4.48 vissen 4.71 ram 4.94 stier 5.24 tweelingen 5.71 kreeft 6.29

Aanvulling op de gebruiksaanwijzing c16:

De c16 berekent uitsluitend goniometrische functies van hoeken die zijn opgegeven in radialen. Om graden om te rekenen naar radialen gebruik je de functie fnr(y) van regel 310 of deze formule: radialen=graden. 11/180.

Wie meer wil weten kan de volgende boeken lezen Sterrengids, uitgegeven door Stichting De Koepel. verschijnt jaarlijks. Practical Astronomy with your Calculator, door Peter Duffet Smith, Cambridge University Press. of de Grundlagen der Ephemeridenrech-

nung, door Oliver Montenbruck, uitgegeven door Sterne und Welteaum, Munshen.

Wij wensen een ieder veel plezier met dit programma en hopen dat de kreet te weinig aandacht voor de c16 even uit de wereld is.

ZON en MAAN

```
20    rem
30    rem Hans Quene
40    rem
50    rem AMSTERDAM
60    rem
100   rem hoofdprogramma
        :
110   gosub310:rem definieren
120   gosub400:rem juliaanse datum
130   gosub470:rem sterrentijd/ecliptica
140   gosub550:rem uitvoer eerste zinnen
150   gosub630:rem eclipticale lengte zo
      n
160   gosub760:rem eclipticale breedte z
      on
170   gosub820:rem eclipticaal-equatoria
      al
180   gosub940:rem equatoriaal-horizonta
      al
190   gosub710:rem afstand aarde zon
200   gosub1110:rem opkomsttijd zon
210   gosub1260:rem ecliptic. lengte maa
      n
220   gosub1400:rem breedte enz. maan
230   gosub820:rem eklipticaal-equatoria
      al
240   gosub860:rem geocentr-topocentrisc
      h
250   gosub940:rem equatoriaal-horizonta
      al
260   gosub1470:rem verlicht oppervlak
270   gosub1520:rem opkomstijd maan
280   ifp$="p"thenprint#4:close4
290   print"[CRSR-DOWN][10xSPACE]<druk[S
      PACE]op[SPACE]toets>":getkeyr$:run
300   rem definieren
        :
310   def fnr(y)=(/180)*(y-(360*int(y/36
      0))):rem graden worden radialen mo
      d 2
320   def fns(y)=atn(y/sqr(-y*y+1)):def
      fnc(y)=/2-fns(y):rem arcsin en arc
      cos
330   dim m$(12)
340   w$(1)="woens":w$(2)="donder":w$(3)
      ="vrij":w$(4)="zater":w$(5)="zon"
350   w$(6)="maan":w$(0)="dins":m$(1)="j
      anuari":m$(2)="februari":m$(3)="ma
      art"
360   m$(4)="april":m$(5)="mei":m$(6)="j
      uni":m$(7)="juli":m$(8)="augustus"
370   m$(9)="september":m$(10)="oktober"
      :m$(11)="november":m$(12)="decembe
      r"
380   return
390   rem juliaanse datum
        :
400   print"[SHIFT-CLR][CRSR-DOWN][10xSP
      ACE]zon[SPACE]en[SPACE]maan"
410   input"[CRSR-DOWN]jaar[2xSPACE]";j:
      ifj<1583thengoto410
420   input"maand[SPACE]";ma:input"dag[3
      xSPACE]";d:input"[CRSR-DOWN]wereld
      tijd";wt:input"minuten[3xSPACE]";m
      n
430   ifma>2theny=j:m=ma:elsey=j-1:m=ma+
      12
440   ew=int(y/400)-int(y/100):ut=wt+mn/
      60
450   jd=int(365.25*y)+int(30.6001*(m+1)
      )+ew+1720996.5+d+(ut/24):return
460   rem ecliptica en sterrentijd
        :
470   input"[CRSR-DOWN]geo[SPACE]breedte
      ";br:gb=fnr(br)
480   input"geo[SPACE]lengte[SPACE]";le:
      gl=fnr(le)
490   print"[CRSR-DOWN]return[SPACE]=bee
      ldscherm":print"p[SPACE]+[SPACE]re
      turn[SPACE]=printen":inputp$
500   ifp$="p"thenopen4,4:cmd4
510   t=(jd-2451545)/36525:e=23.439291-(
      .013004*t):e=fnr(e):t=(jd-2415020)
      /36525
520   ts=6.656306+.0657098242*(int(jd-ut
      /24)-2445700)+(1.0027379093*ut)-(l
      e/15)
530   ts=fnr(ts*15):return
540   rem uitvoer eerste zinnen
        :
550   print"[SHIFT-CLR]":wd=int(jd-.5)-7
      *int(int(jd-.5)/7):pudef"0"
560   print"gevevens[SPACE]van[SPACE]";w
      $(wd);"dag";d;m$(ma);j
570   print"om";wt;"[CRSR-LEFT].";:print
      using"##";mn;:print"[SPACE]wereldt
      ijd[SPACE]op";abs(br);
580   ifbr<0thenprint"zb";:elseprint"nb"
      ;
590   printabs(le);:ifle<0thenprint"ol":
      elseprint"wl"
600   print"[CRSR-DOWN]-----------------
      [CTRL-9]zon[CTRL-0]---------------
      ----[CRSR-DOWN]"
610   return
620   rem ekliptikale lengte zon
        :
630   dp=(1.882-.016*t)*sin(.999026464+2
      .62270627*t)
640   dp=dp+6.4*sin(4.0350267+.352556509
      *t)+.266*sin(.555014702+2.07594181
      *t)
650   lo=279.6966778+(36000*t)+(.768925*
      t)+(3.025e-4*t*t)
660   lo=lo+5.61111111e-5*sin(5.50825912
      +(15.59102621*t))+(dp/3600)
670   g=358.4758333+35999*t+(.04975*t)-(
      1.5e-4*t*t)+(dp/3600):g=fnr(g)
680   dl=(6910.057-17.240*t)*sin(g)+(72.
      338-.361*t)*sin(2*g)+(1.054*t)*sin
      (3*g)
690   l=lo+(dl/3600):l=fnr(l):return
700   rem afstand aarde zon
        :
710   r=(3.057e-5-1.5e-7*t)+(-7.27412e-3
      +1.814e-5*t)*cos(g)
720   r=r+(-9.138e-5+4.6e-7*t)*cos(2*g)+
      (-1.45e-6)*cos(3*g):rz=10^r
730   print"afstand[SPACE]in[SPACE]ae";s
      pc(13);rz
```

```
740   return
750   rem ekliptikale breedte zon
       :
760   u=11.250889+483202.025150*t:u=fnr(
      u)
770   g2=212.45+(58517.493*t):g2=fnr(g2)
780   g5=225.28+(3034.583*t)+.36111111*s
      in(2.334814207+.69710855*t):g5=fnr
      (g5)
790   b=(-.210*cos(2.649410+3*g2-4*g))-.
      166*cos(4.63385-2*g5+g)+(.576*sin(
      u))
800   b=fnr(b/3600):return
810   rem van ekliptikaal naar equatoria
      al:
820   de=sin(e)*cos(b)*sin(l)+cos(e)*sin
      (b):de=fns(de)
830   kl=(cos(b)*sin(l)*cos(e)-sin(b)*si
      n(e))/cos(de):kl=fns(kl):z(1)=kl:z
      (2)=-kl
840   kl=(cos(b)*cos(l))/cos(de):kl=fnc(
      kl):z(3)=kl:z(4)=-kl
850   gosub1070:kl=zq:return
860   rem van geo- naar topocentr. (maan
      ) :
870   h1=r*cos(de)*sin(kl)-cos(gb)*sin(t
      s)
880   h2=r*cos(de)*cos(kl)-cos(gb)*cos(t
      s)
890   h3=r*sin(de)-sin(gb)
900   kl=atn(h1/h2):ifh2<0thenkl=kl+
910   d1=atn(h3*sin(kl)/h1):ifsgn(d1)<>s
      gn(de)thend1=-d1
920   de=d1:r=h3/sin(de):return
930   rem van equatoriaal naar horizonta
      al:
940   tk=ts-kl
950   h=sin(de)*sin(gb)+cos(de)*cos(tk)*
      cos(gb):h=fns(h)
960   a=(sin(de)*cos(gb)-cos(de)*cos(tk)
      *sin(gb))/cos(h):a=fnc(a):z(1)=a:z
      (2)=-a
970   a=-(cos(de)*sin(tk))/cos(h):a=fns(
      a):z(3)=a:z(4)=-a:gosub1070:a=zq
980   h=h*180/:a=a*180/:pudef"[SPACE]"
990   hg=int(abs(h))*sgn(h):hm=60*(abs(h
      )-int(abs(h)))
1000  print"hoogte";spc(13);:printusing"
      ###";hg;:print"[SPACE]";
1010  ifhg=0andh<0thenprint"[2xCRSR-LEFT
      ]-[SPACE]";
1020  pudef"0":printusing"##";hm:pudef"[
      SPACE]"
1030  print"azimuth";spc(11);:printusing
      "####";int(a);:am=60*(abs(a)-int(a
      bs(a)))
1040  print"[SPACE]";:pudef"0":printusin
      g"##";am
1050  return
1060  rem meeste stemmen gelden
       :
1070  forq=1to4:z(q)=z(q)-(2**int(z(q)/(
      2*))):z(q+4)=int(100*z(q)):next
1080  forq=5to8:forq2=qto8:ifz(q)=z(q2)a
      ndq<>q2thenzq=z(q-4)
1090  nextq2:nextq:return
1100  rem opkomsttijd zon
       :
1110  v1=ts:v2=e:v3=jd:v4=t:lz=l:bz=b:dz
      =de:kz=kl:ut=6:s=-1:print"op";spc(
      18)
1120  gosub450:gosub510:gosub630:gosub81
      0
1130  uh=ts-kl:va=(sin(-.01454441)-sin(g
      b)*sin(de))/(cos(gb)*cos(de))
1140  ifuh>thenuh=uh-(2*):goto1140:elsei
      fuh<-thenuh=uh+2*:goto1140
1150  ifabs(va)>1thenprint"niet":goto121
      0
1160  ta=s*abs(fnc(va)):tb=ut*/12+ta-uh
1170  tb=12*(tb/)+.00833
1180  iftb>24thentb=tb-24:goto1180:elsei
      ftb<0thentb=tb+24:goto1180
1190  pudef"[SPACE]":printusing"##";int(
      tb);:print"[SPACE]";
1200  pudef"0":printusing"##";int(60*(tb
      -int(tb)))
1210  ifs=-1thenut=18:s=1:print"onder";s
      pc(15):goto1120
1220  ts=v1:e=v2:jd=v3:t=v4
1230  print"[CRSR-DOWN]-----------------
      [CTRL-9]maan[CTRL-0]--------------
      ----[CRSR-DOWN]"
1240  return
1250  rem gemiddelde lengten maan
       :
1260  lk=270.434164+(480960*t)+(307.8831
      42*t)-(.001133*t*t):lk=fnr(lk)
1270  mk=296.104608+(477000*t)+(198.8491
      08*t)+(.009192*t*t):mk=fnr(mk)
1280  kn=259.183275-(1800*t)-(134.142008
      *t)+(.002078*t*t):kn=fnr(kn)
1290  lg=279.696678+(36000*t)+(.768925*t
      )+(.000303*t*t):lg=fnr(lg)
1300  mg=358.475833+(35640*t)+(359.04975
      0*t)-(.000150*t*t):mg=fnr(mg)
1310  rem ware ekliptische lengte maan
       :
1320  dl=lk-lg:db=(2*lg)-lk-kn
1330  co=22640*sin(mk)+769*sin(2*mk)+36*
      sin(3*mk)-125*sin(dl)
1340  co=co+2370*sin(2*dl)-668*sin(mg)-4
      12*sin(2*(lk-kn))+212*sin(2*(dl-mk
      ))
1350  co=co+4586*sin(2*dl-mk)+192*sin(2*
      dl+mk)+165*sin(2*dl-mg)
1360  co=co+206*sin(2*dl-mk-mg)-110*sin(
      mk+mg)+148*sin(mk-mg)
1370  l=lk*180/+(co/3600):l=fnr(l)
1380  return
1390  rem ekliptikale breedte enz. maan
       :
1400  pi=3423+187*cos(mk)+10*cos(2*mk)+3
      4*cos(2*dl-mk)+28*cos(2*dl)
1410  pi=pi+3*cos(2*dl+mk):pi=fnr(pi/360
      0):r=1/sin(pi)
1420  ss=.272493*sin(pi):ss=ss*180/
1430  b=18520*sin(l-kn+.001997432*sin(2*
      (lk-kn))+.002622842*sin(mg))
1440  b=b-526*sin(db)+44*sin(db+mk)-31*s
      in(db-mk)-23*sin(db+mg)+11*sin(db-
      mg)
1450  b=b-25*sin(lk-kn-2*mk)+21*sin(lk-k
      n-mk):b=fnr(b/3600):return
1460  rem verlicht oppervlak maan
       :
1470  eg=sin(dz)*sin(de)+cos(dz)*cos(de)
      *cos(kz-kl):eg=fnc(eg)
1480  rz=rz*149597870:r=r*6378.14:zm=sqr
      (rz*rz+r*r-2*r*rz*cos(eg)):r=abs(r
      )
1490  f=.5+(r*r+zm*zm-rz*rz)/(4*r*rz):pr
```

```
     int"verlicht[SPACE]gedeelte";spc(9
     );f
1500 print"schijnbare[SPACE]straal";spc
     (9);ss:print"afstand[SPACE]in[SPAC
     E]km";spc(13);r:return
1510 rem opkomsttijd maan
      :
1520 ut=12:s=-1:print"op";spc(18);
1530 dk=kl:du=ut:ut=tb:it=it+1
1540 gosub450:gosub510:gosub1250:gosub1
     430:gosub810:gosub870:uh=ts-kl
1550 ifuh>thenuh=uh-(2*):goto1550:elsei
     fuh<-thenuh=uh+2*:goto1550
1560 va=(sin(0.002327)-sin(gb)*sin(de))
     /(cos(gb)*cos(de))
1570 ifabs(va)>1thenprint"niet":goto166
     0
1580 ta=s*abs(fnc(va))
1590 ifit=1thendd=.9661:elsedd=1.0027-(
     kl-dk)*(12/)/(ut-du)
1600 tb=ut+(ta-uh)*12/(dd*)
1610 iftb<0ortb>24thenvl=vl+1:ifvl=3the
     nprint"niet*":goto1660
1620 iftb<0thentb=tb+24:goto1620:else:i
     ftb>24thentb=tb-24:goto1620
1630 ifit<3thengoto1530
1640 tb=tb+.008333:pudef"[SPACE]":print
     using"##";int(tb);:pudef"0":print"
     [SPACE]";
1650 printusing"##";int(60*(tb-int(tb))
     )
1660 ifs=-1thens=1:vl=0:print"onder";sp
     c(15);:it=1:ut=12:goto1540
1670 return
```

** EINDE LISTING zon en maan **

regel 10	143	regel 370	63	regel 700	30	regel 1030	31	regel 1360	109
regel 20	54	regel 380	142	regel 710	216	regel 1040	196	regel 1370	85
regel 30	107	regel 390	224	regel 720	219	regel 1050	142	regel 1380	142
regel 40	31	regel 400	40	regel 730	80	regel 1060	84	regel 1390	43
regel 50	104	regel 410	213	regel 740	142	regel 1070	148	regel 1400	170
regel 60	143	regel 420	34	regel 750	234	regel 1080	25	regel 1410	205
regel 100	223	regel 430	211	regel 760	237	regel 1090	218	regel 1420	63
regel 110	195	regel 440	206	regel 770	80	regel 1100	24	regel 1430	163
regel 120	1	regel 450	221	regel 780	242	regel 1110	88	regel 1440	7
regel 130	252	regel 460	56	regel 790	70	regel 1120	67	regel 1450	221
regel 140	184	regel 470	208	regel 800	188	regel 1130	239	regel 1460	251
regel 150	196	regel 480	135	regel 810	51	regel 1140	4	regel 1470	171
regel 160	4	regel 490	43	regel 820	216	regel 1150	70	regel 1480	236
regel 170	112	regel 500	42	regel 830	2	regel 1160	138	regel 1490	252
regel 180	163	regel 510	150	regel 840	248	regel 1170	192	regel 1500	171
regel 190	67	regel 520	58	regel 850	75	regel 1180	144	regel 1510	62
regel 200	104	regel 530	34	regel 860	50	regel 1190	221	regel 1520	61
regel 210	118	regel 540	145	regel 870	186	regel 1200	42	regel 1530	50
regel 220	78	regel 550	62	regel 880	185	regel 1210	205	regel 1540	217
regel 230	120	regel 560	161	regel 890	8	regel 1220	102	regel 1550	14
regel 240	93	regel 570	2	regel 900	77	regel 1230	231	regel 1560	153
regel 250	163	regel 580	20	regel 910	76	regel 1240	142	regel 1570	79
regel 260	55	regel 590	83	regel 920	119	regel 1250	183	regel 1580	29
regel 270	147	regel 600	238	regel 930	83	regel 1260	45	regel 1590	37
regel 280	198	regel 610	142	regel 940	58	regel 1270	68	regel 1600	9
regel 290	218	regel 620	174	regel 950	125	regel 1280	201	regel 1610	147
regel 300	162	regel 630	125	regel 960	108	regel 1290	106	regel 1620	200
regel 310	85	regel 640	43	regel 970	197	regel 1300	254	regel 1630	7
regel 320	7	regel 650	23	regel 980	237	regel 1310	9	regel 1640	137
regel 330	171	regel 660	148	regel 990	194	regel 1320	209	regel 1650	159
regel 340	177	regel 670	162	regel 1000	30	regel 1330	237	regel 1660	128
regel 350	170	regel 680	96	regel 1010	252	regel 1340	26	regel 1670	140
regel 360	204	regel 690	46	regel 1020	212	regel 1350	37		

Duikboot C16

Bij dit spel waant U zich ver onder de zee spiegel. Het spel heeft iets weg van het bekende spel zeeslag. Een uitleg is overbodig omdat in het programma wordt verteld hoe het geheel werkt. Bij de meeste mensen zal het wel bekend zijn. Het is geschreven door Henk Boxma uit Ruinen.

```
1    rem duikboot  c-16
2    rem door  h. boxma
3    rem
4    rem       Ruinen
5    rem
10   scnclr
20   print"[CTRL-9]";tab(15);"duikboot"
30   print"[2xCRSR-RIGHT]u[SPACE]bent[S
     PACE]admiraal[SPACE]van[SPACE]de[S
     PACE]kon.[SPACE]marine."
40   print"[CRSR-DOWN][2xCRSR-RIGHT]in[
     SPACE]de[SPACE]haven[SPACE]zijn[SP
     ACE]3[SPACE]vijandelijke"
50   print"[CRSR-DOWN][2xCRSR-RIGHT]dui
     kboten[SPACE]waargenomen.[SPACE]de
     ze[SPACE]moeten"
60   print"[CRSR-DOWN][2xCRSR-RIGHT]ver
     nietigd[SPACE]worden,[SPACE]omdat[
     SPACE]het"
70   print"[CRSR-DOWN][2xCRSR-RIGHT]ned
     erlandse[SPACE]defensie[SPACE]syst
     eem[SPACE]niet[SPACE]uit"
80   print"[CRSR-DOWN][2xCRSR-RIGHT]mag
     [SPACE]lekken.[SPACE]om[SPACE]deze
     [SPACE]boten[SPACE]te"
90   print"[CRSR-DOWN][2xCRSR-RIGHT]ver
     nietigen[SPACE]kunt[SPACE]u[SPACE]
     een[SPACE]aantal"
100  print"[CRSR-DOWN][2xCRSR-RIGHT]die
     ptebommen[SPACE]gooien.[SPACE]hier
     voor[SPACE]moet[SPACE]u"
110  print"[CRSR-DOWN][2xCRSR-RIGHT]coo
```

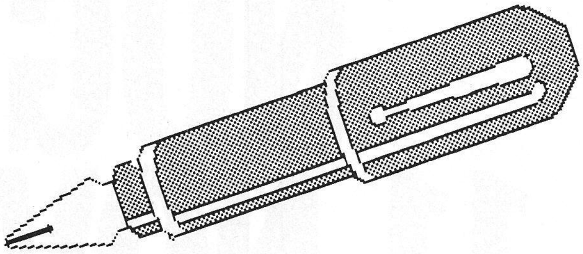
```
    rdinaten[SPACE]van[SPACE]een[SPACE
    ]punt[SPACE]invoeren.[SPACE]de"
120 print"[CRSR-DOWN][2xCRSR-RIGHT]res
    ultaten[SPACE]zijn[SPACE]te[SPACE]
    zien[SPACE]op[SPACE]een[SPACE]kaar
    t."
130 print"[CRSR-DOWN][13xCRSR-RIGHT]sc
    hampschot[SPACE]:[SPACE]#"
140 print"[2xCRSR-RIGHT]symbolen:[2xSP
    ACE]mis[9xSPACE]:[SPACE]Q"
150 print"[13xCRSR-RIGHT]raak[8xSPACE]
    :[SPACE][CTRL-9][SPACE][CTRL-0]";
160 printtab(31)"[CTRL-9]<toets>[CTRL
     0]"
170 vol8
180 restore 700
190 do:readx,y:sound1,x,y*2.0:geta$
200 ifa$=""then210:else220
210 ford=1toy*16+30:next:loopuntilx=0:
    goto180
220 l=0:b=0
230 ee=3:rem aantal schepen
240 dimx(9),y(8)
250 fork=1toee
260 xb(k)=int(rnd(1)*10)
270 yb(k)=int(rnd(1)*9)
280 next
290 z$="[SPACE]0[SPACE]1[SPACE]2[SPACE
    ]3[SPACE]4[SPACE]5[SPACE]6[SPACE]7
    [SPACE]8[SPACE]9"
300 a$="[COM-A]C[COM-R]C[COM-R]C[COM-R
    ]C[COM-R]C[COM-R]C[COM-R]C[COM-R]C
    [COM-R]C[COM-R]C[COM-S]"
310 b$="B[SHIFT-SPACE]B[SPACE]B[SHIFT-
    SPACE]B[SHIFT-SPACE]B[SHIFT-SPACE]
    B[SHIFT-SPACE]B[SHIFT-SPACE]B[SHIF
    T SPACE]B[SHIFT-SPACE]B[SHIFT-SPAC
    E]B"
320 c$="[COM-Q]C[SHIFT-+]C[SHIFT-+]C[S
    HIFT +]C[SHIFT-+]C[SHIFT-+]C[SHIFT
     +]C[SHIFT-+]C[SHIFT-+]C[SHIFT-+]C
    [COM-W]"
330 d$="[COM-Z]C[COM-E]C[COM-E]C[COM-E
    ]C[COM-E]C[COM-E]C[COM-E]C[COM-E]C
    [COM-E]C[COM-E]C[COM-X]"
340 scnclr
350 printchr$(13)
360 printtab(10);z$
370 printtab(10);a$
380 fori=1to8
390 printtab(10);b$
400 printtab(10);c$
410 nexti
420 printtab(10);b$
430 printtab(10);d$
440 print"[HOME][3xCRSR-DOWN]":fori=0t
    o8
450 printtab(8);i;"[CRSR-DOWN]"
460 next
470 print"[HOME][23xCRSR-DOWN]getal[SP
    ACE]voor[SPACE]x:":getkeyx$:x=val(
    x$)
480 print"[HOME][23xCRSR-DOWN]getal[SP
    ACE]voor[SPACE]y:":getkeyy$:y=val(
    y$)
490 ifx>9ory>8then470
500 b=b+1:print"[HOME]beurt";b
510 print"[CTRL-9][CRSR-UP]";tab(15);"
    duikboot"
520 p=3072+11+x*2+y*80+160:pokep,81
530 gosub550
540 goto440
550 rem rekenroutine
560 fork=1toee
570 ifx=xb(k)andy=yb(k)thensound1,100,
    10:sound2,200,10:gosub630
580 ifabs(xb(k)-x)<=1andabs(yb(k)-y)<=
    1thensound3,900,20:gosub610
590 next
600 return
610 p=3072+11+x*2+y*80+160:pokep,35
620 return
630 p=3072+11+x*2+y*80+160:pokep,160
640 xb(k)=-1:yb(k)=-1:l=l+1
650 ifl=eethen670
660 return
670 print"[SPACE][HOME][22xCRSR-DOWN]
    alle[SPACE]schepen[SPACE]zijn[SPAC
    E]tot[SPACE]zinken[SPACE]gebracht[
    SPACE]!"
680 print"[9xSPACE]en[SPACE]dat[SPACE]
    in";b;"beurten."
690 print"[12xSPACE]nog[SPACE]een[SPAC
    E]spel[SPACE]?":getkeya$:ifa$="j"t
    henscnclr:run
700 data 453,10,516,10
710 data 596,20,596,20,685,10,739,10
720 data 770,20,770,30,770,15,739,5
730 data 739,10,704,65,704,10,685,5
740 data 685,60,704,10,685,10,685,10
750 data 643,40,685,10,704,10,596,20
760 data 643,40,453,10,516,10,685,20
770 data 685,60,643,10,596,10,596,80
780 data0,0
```

** EINDE LISTING duikboot **

regel 1	199	regel 130	1	regel 300	153	regel 470	226	regel 640	34
regel 2	176	regel 140	174	regel 310	82	regel 480	230	regel 650	87
regel 3	229	regel 150	142	regel 320	12	regel 490	1	regel 660	142
regel 4	205	regel 160	192	regel 330	159	regel 500	58	regel 670	242
regel 5	130	regel 170	19	regel 340	232	regel 510	205	regel 680	31
regel 10	232	regel 180	35	regel 350	21	regel 520	200	regel 690	2
regel 20	60	regel 190	83	regel 360	127	regel 530	39	regel 700	1
regel 30	62	regel 200	195	regel 370	102	regel 540	33	regel 710	203
regel 40	152	regel 210	57	regel 380	137	regel 550	42	regel 720	148
regel 50	31	regel 220	140	regel 390	103	regel 560	221	regel 730	149
regel 60	106	regel 230	239	regel 400	104	regel 570	169	regel 740	196
regel 70	220	regel 240	118	regel 410	203	regel 580	77	regel 750	190
regel 80	186	regel 250	221	regel 420	103	regel 590	130	regel 760	183
regel 90	136	regel 260	56	regel 430	105	regel 600	142	regel 770	207
regel 100	136	regel 270	17	regel 440	229	regel 610	199	regel 780	14
regel 110	127	regel 280	130	regel 450	177	regel 620	142		
regel 120	81	regel 290	129	regel 460	130	regel 630	246		

Onder de titel GEOS INFO verzorgen Bert Venema en Peter Boncz een rubriek voor de gebruikers van dit populaire besturingssysteem. Lezers worden van harte uitgenodigd om al hun vragen, tips en trucs op te sturen.

GEOS INFO



Applicaties maken

Als u eigen machinetaalprogramma's vanuit GEOS opstart, worden die door GEOS geladen. Vervolgens schakelt de computer kort over naar het standaard C64 openingsbeeld. Direct daarna wordt het machinetaalprogramma met een SYS automatisch vanuit BASIC opgestart. Veel mooier zou het natuurlijk zijn als u uw programma's zou kunnen converteren tot echte GEOS applicaties, die gewoon vanuit GEOS worden opgestart en die na het beeindigen ook automatisch naar GEOS terugkeren.

```
$0460 05 67 65 6F 73 00 66 69  .geos.fi
$0468 6C 65 00 53 61 6D 70 6C  le.Sampl
$0470 65 53 65 71 20 69 6E 66  eSeq inf
$0478 6F 00 63 6C 6F 73 65 00  o.close.
>w
0400 ProgStar      > lda #$C0
0402 ProgStar+$02    sta dispBuff
0404 ProgStar+$04    lda #$04
0406 ProgStar+$06    sta r0H
0408 ProgStar+$08    lda #$28
040A ProgStar+$0A    sta r0L
>s
0402 ProgStar+$02 > sta dispBuff
0404 ProgStar+$04 > lda #$04
0406 ProgStar+$06 > sta r0H
>r
Acc  X   Y    PC   SP  NV-BDIZC  MemMap
$04 $00 $00 $0406 $FD 00100000 00110000
```

De geplaatste machinetaal-listing geeft weer wat u voor uw machinetaalprogramma moet pakken om dit te bewerkstelligen. Overigens moeten machinetaalprogramma's die onder GEOS draaien wel aan een aantal voorwaarden voldoen. Allereerst mag het geheugengebied van $c000 tot $ffff absoluut niet gebruikt worden. Hierin staat namelijk de GEOS kernal. Wordt deze veranderd, dan kan na het beeindigen niet meer worden teruggesprongen naar GEOS. Bovendien moet het machinetaalprogramma, waarvan u een applicatie gaat maken, beschikken over een fileheader.

Dit onderwerp is in de artikelen over GEOS in dit blad al uitgebreid aan de orde gekomen, maar we zullen de belangrijkste dingen nog eens herhalen. De fileheader is een sector op disk. In de directory staat na de naam (dus na 16 tekens) de track en sector van deze fileheader. Overigens staat 2 bytes daar weer na het disktype. Uw programma is voor GEOS een ASSEMBLY programma, dus een 2, dit moet gewijzigd worden in een 6 (de code voor een application). In de fileheader zijn in dit geval alleen de waardes in byte 71-72, 73-74 en 75-76 van belang. Hierin staan resp. het laadadres, het eindadres en het startadres van het programma. Zorg vooral dat deze in orde zijn.

Dan nog dit, er wordt in het programma met een JSR naar de startroutine van uw machinetaalprogramma gesprongen. Als er bij het beeindigen van het programma gebruik wordt gemaakt van een RTS, dan gaat het dus goed. Als het beeindigen van uw machinetaalprogramma anders geregeld is, dan moet u gewoon met een JMP naar $0f4b springen. Als uw programma overigens geen beeindigingsmogelijkheid kent, heeft het natuurlijk weinig zin om er een applicatie van te maken. Tot slot, de machinetaallisting begint op $0f00. U kunt dit echter naar believen wijzigen (zolang het maar niet in Sc000 of hoger komt). Vergeet dan niet de BNE's in S0f17 en S0f2b, alsmede de STA,y in S0f22 te veranderen.

Listing application-maker

```
0f00 jsr Sc232 schakel diskturbo uit.
0f03 jsr Sc25c normale C64 I/O aan.
0f06 lda #S03 begin ROM routine INIT
     SCREEN
0f08 sta S9a EDITOR (normaal $ff81).
0f0a lda #$00
0f0c sta $99
0f0e ldx#$1e
0f10 lda $ecb8,x
0f13 sta $cfff,x
0f16 dex
0f17 bne $0f10
0f19 jsr $e51b
of1c jsr $ff5e
0f1f sei
0f20 ldy #$00
0f22 lda $0f59,y
0f25 sta $0314,y
0f28 iny
0f29 cpy #$20
0f2b bne $0f22
0f2d cli
0f2e jsr $ff8a einde routine.
0f31 jsr $ffcc clear I/O routine.
0f34 lda #$03 juiste bank switch: nr. 3
0f36 sta $dd00(d.i. 0-16384)
     schermkleuren.
0f39 lda #$00
0f3b sta $d020
0f3e lda #$0f
0f40 sta $d021
0f43 lda #$03
0f45 sta $0286
0f48 jsr STARTROUTINE
0f4b jsr $c271 GEOS I/O weer aan.
0f4e lda #$08
0f50 jsr $c2b0 Disk initialiseren.
0f53 jsr $c2a1
0f56 jsr $c22c ga naar DeskTop.
0f59 $31,$ea,$6a,$de,$47,$fe,$4a,$f3,
0f61 $19,$de,$0e,$f2,$0a,$de,$33,$de,
0f69 $57,Sf1,$0e,$de,$ed,$f6,$f1,$12,
0f71 $de,$66,$de,$fe,$02,$de,$66,$de
0f79 $00
```

Wie en wat is Berkeley Softworks?

Met het oog op de naderende Computer Info Beurs (11 & 12 nov.) zijn er vergevorderde onderhandelingen met Berkeley Softworks omtrent het verzorgen van een GEOS seminair. Als alles doorgaat zoals tot nu toe is gepland komt de "grote man" achter Berkeley, Brian Dougherty, naar Nederland om persoonlijk een lezing over GEOS en GEOS-applicaties te houden. In verband hiermee willen we u, in deze rubriek laten kennismaken met de man achter GEOS.

Adviesburo

Brian Doughery is president en hoofdprogrammeur van het in California gevestigde Berkeley Softworks. Opgericht in 1983 functioneerde Berkeley Softworks als een technisch adviesburo. Gedurende 2 jaar ontwikkelde

het bedrijf een variëteit aan produkten voor anderen. Maar ook het ontwikkelen van software voor Apple, Commodore en IBM computers behoorde tot het werkterrein van Berkeley Softworks.
In augustus 1985, begon het bedrijf van Dougherty de ontwikkeling van een nieuw operating systeem voor de Commodore 64, genaamd GEOS. Begin 1986 sloten Berkeley en Commodore een contract , dat GEOS met de nieuwe C64c zou worden meegeleverd. In 1987 werd GEOS onderscheiden met de "Best Technical Achiefment" award, door de Software Publishers Ass. Tegenwoordig is de GEOS produktlijn beschikbaar voor de Commodore 64, 64c en 128, alsmede voor de Apple II-serie. Voordat Berkeley Softworks werd opgericht was Dougherty vice-president bij Imagic, alwaar hij de leiding had bij de ontwikkeling van meer dan 35 software produkten. Dit waren voornamelijk video-games en educatieve software. Tijdens zijn twee-jarige periode bij Imagic, groeide het bedrijf naar een omzet van 80 miljoen dollar per jaar.

Meeting

Dougherty begon zijn carrière bij een afdeling van Mattel Inc., als één van de 4 technici die werkten aan een nieuw produkt genaamd Intellvision. Gedurende zijn medewerking aan het Intellvision systeem, groeide het concept uit naar een 500 miljoen produktlijn met een winst van meer dan 50 miljoen per jaar.
Dougherty behaalde in 1979 aan de Universiteit van California een graad in Electrical Engeneering and Computer Science.

De onderwerpen die tijdens de meeting aan de orde komen zijn nog niet geheel bekend. Maar het is al wel zeker dat de introductie van de nieuwste GEOS upgrade, versie 2.0 tot één van de belangrijkste gespreksitems behoort. Verder treft u op de beurs diverse demonstraties van de meest recente GEOS-applicaties aan. Het is zeker de moeite waard om op één van beide dagen een bezoek aan de GEOS-stand te brengen. Het spreekt vanzelf dat de GEOS-freaks deze ontmoeting met Brian Dougherty niet mogen missen. Dus we zien u op de beurs!!

Bert Venema en Peter Boncz

Voor al uw reakties:
GEOS INFO rubriek
Postbus 43048
1009 ZA Amsterdam

Kleine advertenties

C-128 te koop
C-128, diskdrive 1571, 80-koloms monitor en printer MPS 803: *f* 1300,-.
tel. 02996-3342 (na 18.00 uur)

Diskdrive gevraagd
Te koop gevraagd 1541/C en cartridge expander + module voor C-64.
tel. 02502-8855 (na 19.00 uur)

8032 te koop
Compleet met 8050 dual drive en 8024 printer. Compass boekhoudprogramma, wordcraft tekstverw. *f* 2000,- Tel. 023-263365

Brother HR-5C
printer/plotter compleet met thermisch papier, voeding, kabel, batterijen en handleidingen. Tevens veel programmatuur en boeken voor de C-128 (o.a. robotbesturing en joystick).
tel 01837-1613

C-128 te koop
Compleet met monitor 40-80 koloms, diskdrive, joystick, software en boeken. Prijs *f* 1200,-. Tel. 04192-18679

C-64 systeem
C-64, 1541 drive, Brother HR-5C, 2 joysticks, kleurenmonitor 1702, datarec., boeken, en software (o.a. Geos v1.2). *f* 1500,- Tel. 078-144377

Te koop gevraagd
2e-hands 1541-diskdrive met handleiding voor max. *f* 125,- met event. diskettes (*f* 0,50 p/st). Tel. 02520-19465

1084 monitor te koop
Monitor voor Commodore, nieuw in doos. Compleet van *f* 900,- voor *f* 750,-. Tel. 08851-15293

C-64 te koop
Met diskdrive 1541 C, datarecorder, Power cart., Simon's Basic Cart., joystick, 50 diskettes, cassettes, 800 programma's, Geos v1.2. Prijs *f* 950,- Tel. 05427-14517

C-128 systeem
te koop C-128 met datarec. 1530, 1541 drive, diskettes in originele verpakking en met handleiding.
tel. 010-4342179 (Jan)

Origineel pakket
Te koop in originele verpakking: C-64, diskdrive, datarec., final cart., mini Office II, Geos e.a. Alles in één koop: *f* 1100,-. Tel. 078-158769

Printer te koop
Gen. Electric printer voor C-64/128 en MSX voor ketting, thermisch en normaal papier. Incl. therm. papier A4, lint, handl. Z.g.a.n. Tel. 023-241164

Software te koop
In originele verpakking met handleiding, Flight Deck, A view to Kill, Back to the Future, Oberon, Winter Games, Darkside en vele andere. Prijzen tussen *f* 7,50 en *f* 35,-.Tel. 075-175357

Boekhoudmachine
Te koop NCR 299 electr. boekhoudmachine. Met 2000 kaarten, papierrollen en programma's voor boekhouding en salarisadm. *f* 200,-.
tel. 070-453673

C-128 systeem te koop
Met 40-80 koloms monitor, Superscript, Geos, veel software, NL handleiding, diskettebox en tijdschriften. Alles in originele verpakking.
tel. België: 057-204051 (na. 17.00)

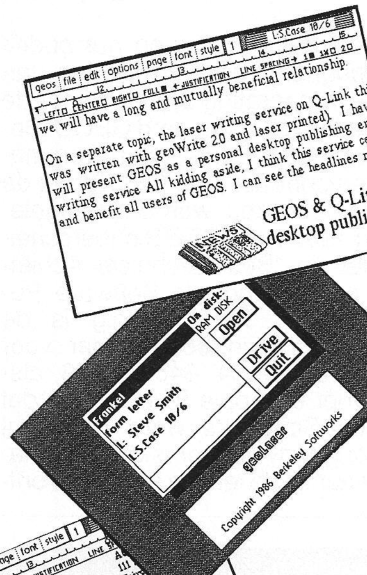
Gezocht
Adventures voor C-64 op tape of disk. In ruil voor andere programma's.
tel. 05981-260/050-261343

C-64 te koop
Met 1541 diskdrive, datarec., final cartr., 2 joysticks, software, handl. *f* 800,- Tel. 01820-33870

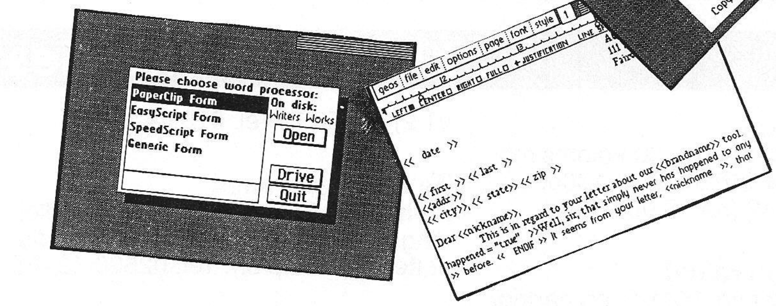
Het zat al enige tijd in de lucht. De nieuwste, sterk verbeterde versie van de grafische interface GEOS is een feit. Na enige geruchten, uit voornamelijk de buitenlandse pers is de versie 2.0 van dit populaire pakket nu op de markt gebracht.

GEOS 2.0

.... de tweede generatie !!



Dat, ondanks negatieve klanken betreffende de levensvatbaarheid van C-64 software, er toch nog gerenomeerde ontwikkelaars zijn, die zich daaraan niet storen wordt nu weer eens bewezen door Berkeley Softworks. Daar zitten de programmeurs van het grafische pakket GEOS, dat nu aan zijn "tweede leven" gaat beginnen. Nog sneller, geavanceerder en slimmer dan de vorige versie 1.3, waarover u in dit artikel alles kunt lezen.



De nieuwe versie van GEOS, welke is gedoopt als versie 2.0, bestaat uit een gewichtige samenstelling. Een drietal, dubbelzijdige diskettes en een 307 pagina tellende handleiding vormen de inhoud van de doos, die voor deze gelegenheid ook in een nieuw jasje is gestoken. De hoeveelheid diskettes geeft al enigszins aan dat er het een en ander is uitgebreid t.o.v. de vorige versie. Naast een aantal "nieuwe" applicaties hebben de reeds bestaande, standaard tot het pakket behorende applicaties, vele verbeteringen ondergaan. De handleiding is opgezet in 9 hoofdstukken, die elk afzonderlijk een applicatie beschrijven. De eerste 2 hoofdstukken vertellen de gebruiker hoe hij aan de slag kan met GEOS en hoe er mee om te gaan. Een aantal appendices vormen het laatste hoofdstuk.

Installeren

Zoals alle software die door Berkeley wordt uitgebracht, moet ook GEOS 2.0 geïnstalleerd worden. Dit installeren is bij een applicatie tot doel om illegale copieën onbruikbaar te maken. Bij GEOS 2.0 is dit installatieproces bedoeld om het pakket voor alledaags gebruik gereed te maken. Maar het belangrijkste bij dit installeren is wel de mogelijkheid om reeds eerder gebruikte applicaties, zoals Deskpack 1, GeoCalc, GeoFile of GeoPublish met deze nieuwe versie te kunnen "booten". Dit betekent dus, dat als u reeds enige van deze applicaties bezit en destijds heeft geïnstalleerd met GEOS 1.3, deze nu ook kunt gebruiken met de nieuwste GEOS versie. Er is één "maar" aan verbonden, u dient wel over *orginele* diskettes te beschikken. Anders gaat het installeren van deze mogelijkheid niet goed en kunt u alleen uw oude 1.3 versie gebruiken als "bootdisk". Na het installeren is het een goede zaak om een copie van de diskettes te maken. Dit is mogelijk met alle applicaties met uitzondering van de "bootdisk", welke is beschermd tegen copieëren. Maar u kunt gerust zijn, want de leverancier heeft van de "bootdisk" een back-up meegeleverd.

Benodigde hardware

GEOS 2.0 draait moeiteloos op een Commodore 64/128 met een 1541 diskdrive. Maar u heeft wel meer profijt van het pakket, wanneer u gebruik maakt van een Commodore 128, een RAM Expansie Unit (REU) en een tweede diskdrive (bv. 1571). GEOS 2.0 ondersteunt nu zelfs het gebruik van een 1581 drive (3,5 inch), hetgeen de opslagcapaciteit zeer ten goede komt.

U geeft deze specificaties met betrekking tot de te gebruiken hardware op in het "configure"-file. Nadeel van dit file is dat het op iedere werkdiskette moet staan die u gaat gebruiken. Steeds wanneer u een applicatie beëindigd zoekt GEOS naar dit file. Indien dit "configure"-file niet op de disk staat, die op dat moment wordt gebruikt, zal GEOS één diskdrive herkennen.
Voordeel van het gebruik van een 1571 of 1581 diskdrive is dat ze dubbelzijdig kunnen worden geformateerd, zelfs wanneer u een C64 gebruikt. Vooral bij het gebruik van GeoWrite, GeoPaint en GeoPublish is dit een welkome verbetering. De geïnstalleerde configuratie wordt altijd op de DeskTop weergegeven (zie fig. 1). Het gebruik van 2 drives en een RUE is nu eveneens mogelijk.

De desktop

Het eerste wat opvalt van de vernieuwde DeskTop is de uitgebreide menu-balk, bovenin. Twee nieuwe pull-down menu's zijn hierbij gemaakt. De eerste, "Select", heeft betrekking op selectie van files, waarbij meervoudige selecties op een pagina of in een border mogelijk is. De optie "page" biedt de mogelijkheid om een DeskTop pagina toe te voegen of te verwijderen.

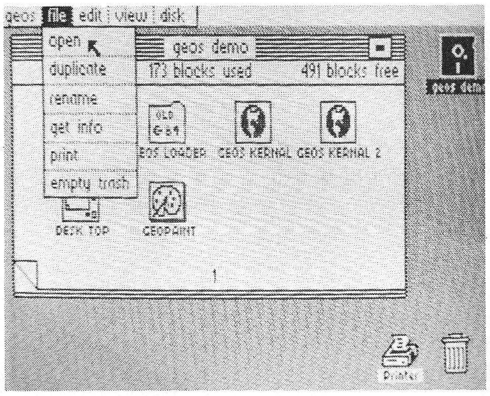
Andere uitbreidingen in de menu-balk treffen we aan wanneer we een optie selecteren. Onder de optie GEOS treffen we de keuze "pad color mgr" aan (zie fig. 2), een desk-accessoire om bepaalde filetypen eenzelfde kleur te geven, alsmede het kleuren van de voorgrond en achtergrond. Met deze functie in combinatie met de preference manager kunnen we stellen dat GEOS 2.0 nu geheel in kleur werkt.
Met de optie "undo delete" uit het "file"-menu kunnen we het laatst verwijderde file weer terughalen in de DeskTop. De naam van zo'n verwijderd file staat altijd onder de prullemand weergegeven, tenzij het "validate" commando is gebruikt.
Nieuw is ook de constante aanwezigheid van de datum en tijd in de rechterbovenhoek van het scherm. De actuele datum en tijd hoeven, in tegenstelling met de vorige versie 1.3, niet meer opgegeven te worden middels de preference manager, maar kunnen d.m.v. de optie "set clock" uit het "options"-menu worden ingesteld.

De applicaties

De standaard samenstelling van het GEOS 2.0 pakket is ook behoorlijk uitgebreid, hetgeen aan de prijs niet echt te merken is. In Amerika is het pakket slechts $10,00 duurder. Wat de prijs hier gaat worden is nu nog niet bekend. U krijgt er in ieder geval een veel uitgebreider geheel voor. Zo treffen we op een van de diskettes de nieuwste versie 2.1 van GeoWrite aan, een verbeterde GeoPaint, een sterk versnelde DiskTurbo en de standaard desk-accessoires, zoals de calculator, het 127 pagina tellende notepad, een alarm-klok, photo- en textalbums en photo- en textmanagers. Nieuwe programma's die worden meegeleverd zijn o.a. de applicatie Paintdrivers, om GeoWrite documenten te converteren naar GeoPaint, GeoSpell, de spelling-checker, TextGrabber, voor tekstbestanden conversie, alsook GeoMerge, om mail-merge files te creëeren, behoren tot de standaarduitrusting.
Verder wordt GEOS 2.0 geleverd met 31 printerdrivers, die ervoor zorgen dat zo'n 70 populaire printers kunnen worden aangestuurd. Ook voor de bezitters van een LaserJet printer is er een GeoLaser driver meegeleverd.

Geowrite 2.1

Een professionele tekstverwerker die mogelijkheden bezit als word-wrap, proportioneel letterschrift, individuele paragraaf formatering, verschillende tab-instellingen (waaronder decimale tabs) en verschillende opties voor regel spatiëring. Daarnaast biedt GeoWrite 2.1 de mogelijkheid om graphics uit GeoPaint te importeren en op een willekeurige plaats in het dokument te plaatsen, zodat tekst om het plaatje wordt geplaatst. Het instellen van pagina-breeklijnen, de volle 8-inch werkbreedte, links, rechts of gecentreerd positioneren van tekst behoren tot de vele mogelijkheden.
Invoegen, verwijderen, verplaatsen en copiëren van tekst is kinderspel met GeoWrite 2.1.
Wanneer men gebruik wil maken van kop- en voetteksten, is dit mogelijk tot zelfs 1/3 van de paginalengte. Zoeken en vervangen van teksten, woorden of zelfs tekens is in GeoWrite simpel te bewerkstelligen. Tegelijkertijd kan men 10 verschillende letterfonts in 7 verschillende stijlen gebruiken in een dokument (zie fig. 3). Wanneer men dubbel klikt met de muis op een woord dan wordt deze automatisch geselecteerd. Datum en tijd kunnen eenvoudig worden ingevoerd met één opdracht. En tot slot kan men het uiteindelijke dokument geheel of gedeeltelijk, in draft, NLQ of high quality afdrukken.



Paint drivers

Deze applicatie wordt gebruikt om speciale effecten te creëeren met een GeoWrite dokument. Met het Paintdriver programma ontwerpt u een minikrant, voorzien van kolommen en tekeningen. Of u maakt een mooie koptekst en randvoorzieningen met graphics uit GeoPaint of Graphics Grabber.

Geospell

Nieuw is ook de applicatie GeoSpell, waarmee de gebruiker snel verkeerd gespelde woorden kan opzoeken en verbeteren. De bibliotheek GeoDictionary) van GeoSpell bevat zo'n 38.000 woorden, helaas nu nog alleen in het Engels. Maar de applicatie biedt de mogelijkheid om persoonlijke bibliotheken aan te leggen. Deze versie van GeoSpell is zo'n 38% sneller dan de vorige, separaat geleverde versie. (zie fig. 4).

Geopaint

Met GeoPaint maakt u "high-resolution" tekeningen met behulp van een volledig uitgeruste "toolkit". Diagrammen, graphics kunnen tot een grootte van 8 bij 11 inch worden ontworpen. Uit vergroten van een detail om m.b.v. "pixel-edit" (puntje voor puntje) de tekening te preciseren verhoogt de nauwkeurigheid.
GeoPaint bezit 16 kleuren, 14 tekentools, 32 kwasttypen en 32 invulpatronen. Tekeningen kunnen worden geïnverteerd, geroteerd, verplaatst of worden gespiegeld. Nieuw in GeoPaint is de mogelijkheid aaneengesloten lijnen in figuren te tekenen. Het vervaardigen van cirkels en ellipsen is één van de mogelijkheden, alsmede het uitvergroten en op schaal invoeren van "photo-scraps". Om "overlay" effecten te bereiken kan men gebruik maken van de transparant-modus (zie fig. 5). Instellen van de "grid" lijnen maakt het mogelijk om met het meegeleverde sjabloon tekeningen en plaatjes zeer nauwkeurig na te tekenen.
Kortom, GeoPaint is een tekenprogramma met veel allure (zie fig. 6).

Desk accessoires

Het is nu ook mogelijk om het resultaat van een berekening met de "calculator" automatisch in te voeren in een ander dokument. De calculator kan in elke gewenste applicatie als "pop-up" worden gebruikt om even vlug wat rekenwerk te doen. Ook het transporteren van gegevens uit de NotePad is een vernieuwing. Zo ook

het benoemen van plaatjes en tekststukjes, die men heeft verzameld in resp. het Photo-Album en Text-Album. Zoeken naar een bepaald plaatje of tekst behoort nu ook tot standaard functies van deze laatste twee accessoires.

Text grabber & Geomerge

Voor het importeren van niet GEOS georiënteerde tekstdokumenten, die afkomstig zijn van andere word-processors, gebruikt men TextGrabber. Hiermee importeert men op eenvoudige wijze files, welke zijn vervaardigd met PaperClip, EasyScript, SpeedScript, WordWriter of een andere Commodore tekstverwerker.
Met GeoMerge kunnen mail-merge files worden vervaardigd. Hiermee wordt bedoeld dat men een persoonlijke brief maakt met GeoWrite en dan m.b.v. GemoMerge de bijbehorende gegevens combineert met die brief. Die gegevens kunnen afkomstig zijn uit een GeoFile- of GeoDex-bestand, maar net zo goed uit een ander GeoWrite dokument.

Algemeen

Men kan tijdens het werken met GEOS het beste gebruik maken van Commodore 1351 mouse, maar als u daarover niet beschikt, is het gebruik van het toetsenbord ook mogelijk. Meer dan voorheen bevat GEOS 2.0 short-cuts. Dit zijn enkelvoudige- of dubbele toetsaanslagen om een functie te activeren. De DiskTurbo is met het lezen en schrijven naar diskette zo'n 7 maal sneller geworden. Diverse "input-drivers" worden meegeleverd, zoals joystick-, lichtpen-, koala paden proportionele mousedrivers.
GEOS 2.0 bevat tevens een uitgebreide demonstratie-diskette, die de GEOS gebruiker in staat stelt kennis te maken met andere GEOS applicaties zoals GeoPublish, GeoFile, GeoCalc, Deskpack Plus en Fontpack Plus.

Tot slot

We kunnen gerust stellen dat Berkeley Softworks erin geslaagd is een goede vervanger van de oude 1.3 versie te vervaardigen. Veel nieuwe programma's en sterke verbeteringen behoren tot de standaard-uitrusting van GEOS 2.0. Ook voor de huidige gebruikers een aanrader om aan te schaffen, zeker wanneer men van plan is de computer uit te breiden met een REU en/of tweede diskdrive. Met name de snelheid en de nog meer verbeterde gebruikersvriendelijkheid van GEOS 2.0 springen eruit. In de aan mij ter beschikking gestelde versie van GEOS 2.0 (vers van de pers) zat echter nog een kleine "bug", die wellicht in de commerciële versie zal zijn verdwenen. Wanneer u uzelf wilt overtuigen van de capaciteiten die dit pakket bezit, is het verstandig de komende Computer Info Beurs op 11 en 12 november te bezoeken. Daar worden demonstraties met de nieuwe GEOS 2.0, ook lezingen, over GEOS en de aanverwante applicaties gehouden.
We kunnen met een zekere arrogantie wel zeggen:
"GEOS HAALT MEER UIT UW COMMODORE DAN MEN ZOU VERWACHTEN!"

Bert Venema

Kleine advertenties

Te koop C-64
c-64, d.rec, final cartr.II, 2 joysticks, PBNA cursus, z/w TV, veel software, alles 1 jaar oud, voor *f* 700,-.
tel. 05750-41412 (na 18.00 uur)

Brother HR 5 C
Matrixprinter geheel compleet, incl. papier, adapter en inkt. *f* 250,-
tel. 053-333140 (na 18.00 uur)

C-64 II te koop
Compleet met 1541 diskdrive, 1530 datarecorder, KCS, joystick, boeken, 50 diskettes in orig. verpakking, kl.monitor, diskbox, stofhoes. Prijs *f*1300,-
Tel. 070-893751

Help!
Met mijn printer MPS 802 werken grafische programma's (Doodle) niet. Wie heeft hulpprogramma of weet raad? Tel. 04132-64239 (na 17.00)

Star SG 10C
Weinig gebruikte Star printer aangeboden. Direkt aansluitbaar op Commodore *f* 200,- Tel. 020-652994 (Eric)

Printer te koop
C.Itoh model 8510, van *f*2400,- voor *f* 250,- Tel. 02280-13097 (na 17.00)

Totaal systeem te koop
C-64, diskdrive en monitor, incl. disks, handleiding en user manual totaal *f* 800,-. excl. extra MTS 802 printer.
tel. 015-126781/123297

Commodore 1520
Te koop z.g.a.n. 1520 colour-printer/plotter + extra's. *f* 150,-
tel. 033-723822 (na 18.00 uur)

Philips monitor te koop
Groen, voor C-128 40-80 koloms. Prijs n.o.t.k. Tel. België: 031669.85.50

Te koop C-128
incl. monitor, datarec., joystick, MPS 803, IBM thermo-printer, diskdrive 1570, super Toolkit, boeken, software, *f* 1700,- Tel. 02510-20961

Commodore C-16
Te koop C-16, datarecorder, joystick, spelletjes, NL & Eng. handl., cursus. prijs *f* 200,- Tel. 01740-29335

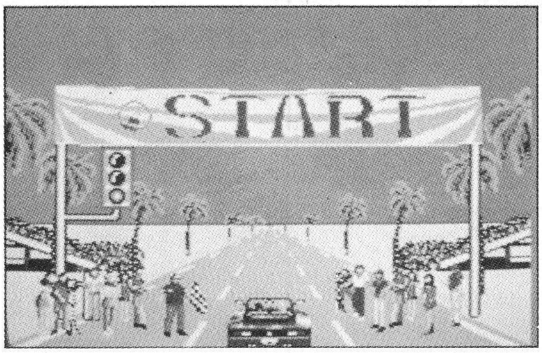
Printer gevraagd
Matrixprinter voor C-64 gevraagd en diskdrive. Wel graag in goede conditie. tel. 03210-14082 (na 18.00 uur)

Power Cartridge
Te koop Power cartridge II met orig. handl. Prijs *f* 65,- Tel. 02159-15823

Gevraagd
Power cartridge copie op tape voor besturing van de machinetaal.
tel. 01823-4560 (Ron)

Het opwindende gevoel van Out Run

Een ware sensatie op het gebied van racesimulatoren is het arcadespel Out Run. Het kostte de programmeurs zo'n 9 maanden om het grafisch ontwerp en spelelement om te zetten in een computerspel. Van oorsprong wordt dit spel, bekend geworden in de video-speelhal, gespeeld in een nagebouwde cockpit van een Ferrari Testarossa. Verkrijgbaar in verschillende computerversies, waaronder de C64/C128, wordt Out Run geleverd voor een prijs die rond de 35 gld. ligt. Tevens wordt de originele muziek op een aparte audiocassette meegeleverd.



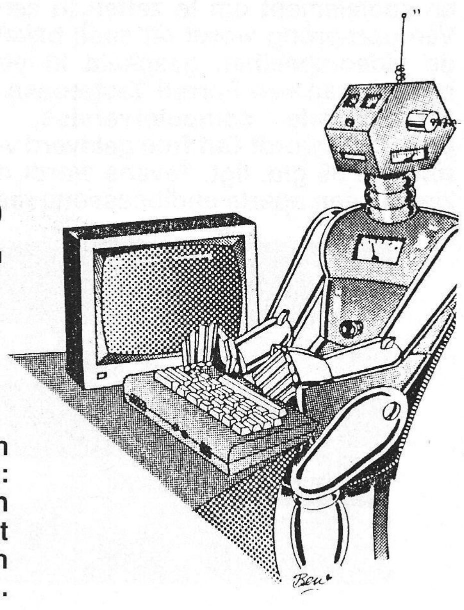
De bedoeling van het spel is, om binnen de beschikbare tijd (en die is niet erg lang) een bepaald parcours af te leggen. Vanzelfsprekend dient hierbij gebruik gemaakt te worden van de 5 versnelling tellende, 4942 cc motor van de Ferrari, die met een snelheid van maximaal 180 mph. over de weg raast. Er zijn vijf trajecten, waaruit de coureur kan kiezen. Je start de motor, grijpt de handles en neemt deel aan de snelste race ooit verreden. Allerlei obstakels kom je onderweg tegen, hetgeen een beroep doet op je rijvaardigheidskunsten. Scherpe bochten, medeweggebruikers, bomen, heuvels en nog veel meer, kunnen het af te leggen traject bemoeilijken, zodat de tijdslimiet nauwelijks gehaald kan worden. Een juist gebruik van de gaspedaal en versnellingspook voert je door het landschap, dat schitterend wordt weergegeven. Elke scene moet binnen de gestelde tijd worden doorkruist. Elke route bestaat op zich weer uit 5 scenes. Zo rij je op een zeker moment door Devils Canyon, de wildernis, een wijngaard, of door een graanveld, over een bewolkte berg, om uiteindelijk één van de eindbestemmingen te bereiken. Alle acties worden bestuurd met behulp van de joystick, waarbij voor- of achteruit zorgt voor resp. versnelling en afremmen. De vuurknop zorgt voor het schakelen naar een hogere versnelling. Het spel is ten einde op het moment dat de beschikbare tijd is verstreken of wanneer men het einde van het parcours heeft bereikt.

Het spel vergt op alle niveau's een zeer behendig gebruik van de joystick. Het is zeker geen makkelijk spelletje, dat je zo maar even speelt. Eindelijk weer eens een echt arcadespel, waarvan je terecht kunt zeggen dat het uiterste van je reactie- en coördinatievermogen opeist. Het spel is binnenkort ook leverbaar voor de Amiga. De C64-versie is leverbaar op cassette en diskette, voorzien van een korte, maar duidelijke Engelse handleiding. Een must voor iedere snelheidsduivel.

Bert Venema

In dit tweede deel van machinetaalprogrammeren in GEOS behandelt Peter Boncz het maken van menu's en icons, het gebruik van GEOS disk-routines en speciale grafische routines. Deze onderwerpen worden ingepast in een handleiding voor het schrijven van een eigen applicatie. De bijgeplaatste listing is die testapplicatie: uw eerste eigen machinetaalprogramma in GEOS!

GEOS Cursus machinetaal: deel 2



Het eerste GEOS-programma.

De vorige aflevering bevatte de introduktie voor het programmeren in GEOS. De toen verstrekte gegevens worden nu in de praktijk toegepast: we gaan zelf programmeren! Vorige maand werd ook een schema met een volledig overzicht van alle GEOS-routines belooft, wegens plaatsgebrek zult u daar nog even op moeten wachten. Dit is niet erg, want alle routines die in deze cursus gebruikt worden, zullen steeds van tekst en uitleg voorzien zijn.

Maar eerst gaan we nog eens terug naar wat we de vorige keer hebben behandeld. Machinetaal-programmeren in GEOS is het makkelijkst met de GeoAssembler. Deze GeoAssembler gebruikt het systeem van labelen (zoals alle assemblers). Variabelen en routines, die door GEOS gebruikt worden, zijn gelabeld: ze hebben een naam gekregen. Verder kent de GeoAssembler functies als:
'.byte WAARDE (0-255)', '.wordt WAARDE (0-65535)' en '.psect ADRES'.
In de vorige aflevering zijn deze speciale functies al behandeld. Alles wat er in deze cursus besproken zal worden, zal echter ook met gewone machinetaalmonitoren gedaan kunnen worden (dus zonder die speciale functies). Het merendeel van de lezers bezit namelijk geen GeoAssembler.

Programmastructuur

Een GEOS programma bestaat slechts uit een initialisatieroutine, wat tabellen (om menu's en iconen te definiëren) en uitvoerroutines. Als het programma loopt, wordt er van deze routines geen gebruik gemaakt. GEOS bevindt zich dan in MAINLOOP (het hoofdprogramma) of INTERRUPTVECTOR (de rasterinterrupt). Alleen aan het begin wordt de initialisatieroutine doorlopen. Verder komt de doorsnee applicatie slechts aan bod als er een actie moet worden ondernomen, bijvoorbeeld als een menu is geselecteerd. In de listing van de TESTAPP loopt de initialisatieroutine van $9ffc tot $1021.

De grootste kracht van het GEOS-systeem ligt in de vele hapklare routines die de kernel biedt. De programmeur hoeft niet iedere keer weer het wiel uit te vinden en heeft al standaard een heleboel routines ter beschikking voor bijv. vermenigvuldigen, delen, maar ook het trekken van een HIRES lijn etc. Zo'n routine haalt dan vaak zijn informatie uit speciale GEOS-tabellen die de programmeur in het geheugen moet zetten. Het leren programmeren in GEOS bestaat dus vooral uit het goed leren omgaan met deze al bestaande faciliteiten. Ik zal het dus vooral over GEOS-routines en tabellen hebben in deze cursus.

Routines

De eerste routine die ik zal bespreken is jsr DoMenu. Deze functie zet een menu op het scherm en kijkt daarvoor in register 0 (R0). Zoals vorige keer is uitgelegd worden hier de geheugenplaatsen $0002 en $0003 mee bedoeld. Deze twee adressen moeten pointen naar de menu-tabel. Als bijvoorbeeld de menu-tabel op $1000 staat, moet $0002 de waarde $00 hebben en $0003 moet de waarde $10 hebben. Voordat dus jsr DoMenu gedaan wordt, moeten dus eerst deze twee geheugenplaatsen van de juiste waarden voorzien worden. Dit is ook te zien in illustratie 1.

MENUS & ICONS

naam routine	adres	variabelen in	vernietigd	omschrijving
DoMenu	$c151	a -nummer keuze waar pijltje komt, r0 -pointer naar menu-tabel.	r0-r13,a,x,y	zet het menu op en activeer het.
GotoFirstMenu	$c1bd	geen.	r0-r13,a,x,y	Rol submenus op.
DoIcons	$c15a	r0 -pointer naar icon-tabel.	r0-r11,a,x,y	zet een ikoon op

Illustratie 1

Onder het kopje 'variabelen in' staat welke variabelen in het geheugen van tevoren van een waarde voorzien moeten worden en waarom. Onder het kopje 'vernietigd' staan alle variabelen die door de routine zelf veranderd worden. Zet hier dus geen informatie in als U de desbetreffende routine aanroept!

Menu tabel

De adressen $0002 en $0003 wijzen dus naar de Menu-tabel. Deze tabel bepaalt volledig hoe een menu eruit gaat zien. In GEOS zijn er 2 soorten menu's mogelijk: horizontale en verti-

cale. De menu's die constant op het scherm aanwezig zijn worden hoofdmenu's genoemd. Meestal zijn die horizontaal. Wordt zo'n menu aangeroepen dan verschijnt er een doorgaans verticaal SUBmenu (pull-down menu)

```
                 Menu Tabel
.byte y coordinaat bovenkant    (0-199)
.byte y coordinaat onderkant    (0-199)
.word x coordinaat linkerkant   (0-319)
.word x coordinaat rechterkant  (0-319)
.byte menutype geORd met aantal keuzes.
      bit 7 aan: vertikaal
            uit: horizontaal
      bit 6 aan: pijltje kan alleen in
                 menu bewegen.
            uit: pijltje kan overal
                 bewegen.
      bit 5-0  : aantal keuzes.
.word pointer naar tekst keuze 1.
.byte keuzetype
      bit 7 aan: pointer hierna wijst naar
                 submenu.
      bit 6 aan: pointer hierna wijst naar
                 dynamic submenu.
      bit 7 en bit 7 uit: pointer hierna
                 wijst naar routine.
      bit 5-0:   ongebruikt.
.word pointer submenu/routine (hangt dus
      van keuzetype af).
.word pointer naar tekst keuze 2.
.byte keuzetype
.word pointer submenu/routine (hangt van
      keuzetype af).
etc. (voor zoveel keuzes als opgegeven bij
      menutype).

Een routine moet beindigd worden met een RTS
De structuur van een submenu is precies het-
zelfde als het schema hierboven.
```

Illustratie 2

op het scherm. Het kan echter ook zijn dat de computer onmiddellijk actie onderneemt wanneer een horizontaal menu geselecteerd is.
In illustratie 2 vindt U de structuur van de Menu-tabel. Als de werking U niet geheel duidelijk is, kunt U ook naar de listing van de TESTAPP kijken. De Menu-tabel begint daar in $1022. In $1022 tot $1027 staan de coördinaten van het menu. De geheugenplaats $1028 bevat het menutype (dit is $00-dus horizontaal) geORed met het aantal keuzes (2). In $1029 en $102a staat dan het adres van de tekst die in het eerste menu-hokje staat. Als U de uitleg goed leest, dan wijst het zich allemaal vanzelf.

Icons

Als het principe van de menus u duidelijk is, zullen de Icons niet veel moeilijkheden geven. Een Icon is in GEOS een plaatje dat een functie aanduidt. Klikt men het Icon met de muis aan, dan wordt er een routine gestart.
De functie jsr DoIcons zet de Iconen op het scherm en activeert ze. Net als bij Domenu wijst r0 naar de Icon-tabel. De structuur van deze tabel staat in illustratie 3.

Diskroutines

In deze aflevering bespreek ik ook een aantal diskroutines. GEOS kent verschillende niveaus voor het gebruik van diskroutines.
De routines die deze keer aan bod komen zijn van het hoogste niveau. Hoe lager het niveau, hoe meer kleine zaken de programmeur zelf moet regelen. Deze 'high-level' routines zoeken alles uit voor de programmeur, deze kan in zijn luie leunstoel achterover geleund blijven liggen.
Even in een notedop: Opendisk leest de gegevens in van de disk die in de drive zit. Newdisk doet iets soortgelijks, SetDevice geeft de IO-poort aan het devicenummer dat in de accu (dus lda #devicenummer) staat. Getfile laadt een file in. De naam van de te laden file staat in het adres waarnaar in in R6 ($000e,$000f) gepoint wordt. Deze naam wordt beëindigd met een #$00. Is de file een applicatie, dan wordt deze na het laden ook gerund. De computer keert dan niet terug naar de plaats waar jsr GetFile stond. Als de file een Desk Accessory (bijv. de rekenmachine) is, wordt deze ook gerund, maar de computer keert wel terug naar de jsr GetFile.
Tenslotte is er nog EnterDesktop, deze routine start de DESKTOP, en keert nooit naar jsr $EnterDesktop terug. Een overzicht van deze routines staat in illustratie 4.

Grafische routines

Tot slot behandel ik enkele eenvoudige grafische routines. Zoals vorige keer al besproken werd heeft GEOS een voorgrond- en een achtergrondscherm. De geheugenplaats $002f, die in GeoAssembly DisplayBufferOn heet, bevat informatie hierover. Als bit 7 aanstaat, is het voorgrondscherm in gebruik. Als bit 6 aanstaat, kan er ook naar het achtergrondscherm geschreven worden. Het achtergrondscherm wordt gebruikt als buffer om de stukken scherm, die achter een submenu of dialogbox stonden, te zijner tijd weer tevoorschijn te halen. Normaal staan beide bits aan, alleen als u echt hele grote programma's gaat schrijven kan het zijn dat u geen achtergrondscherm wilt hebben. Op dit onwaarschijnlijke geval zal ik niet ingaan.

Dan nu ter zake. In illustratie 5 staat een schema met grafische routines. De functies van HorizontalLine en VerticalLine spreken voor zich. SetPattern wordt veel gebruikt in combinatie met Rectangle. Met Setpattern wordt een patroon geselecteerd, waarmee met Rectangle een rechthoek wordt opgevuld. i_Rectangle is de 'inline' versie van Rectangle en

DISK

naam routine	adres	variabelen in	vernietigd	omschrijving
Opendisk	$c2a1	geen.	a,x,y,r0-r15	
NewDisk	$c1e1	geen.	a,x,y,r0-r15	
SetDevice	$c2b0	a -device nummer.	a,x,y.	
GetFile	$c208	r6 -point naar filenaam, r0L-optie.	a,y,r0-r10.	
EnterDesktop	$c22c	geen.		komt niet terug
MouseUp	$c18a	geen.		pijltje aan.

Illustratie 4

```
                     Icon Tabel
.byte aantal icons
.word x coordinaat waar pijltje komt na het neerzetten v.d. icons.
.byte y coordinaat waar pijltje komt na het neerzetten v.d. icons.

.word pointer naar grafische data van icon 1.
.byte x coordinaat in bytes (*8 dus) v.d. linkerbovenhoek icon 1.
.byte y coordinaat in pixels v.d. linkerbovenhoek icon 1.
.byte breedte icon 1 in bytes (*8).
.byte hoogte icon 1 in pixels.
.word routine om naar te springen als icon 1 geselecteerd is.

.word pointer naar grafische data van icon 2.
etc.
De grafische data van een ikoon bestaat uit breedte*hoogte
bytes, afgesloten met een $00.
De routines waar eventueel heengesprongen wordt, moeten afgesloten
worden met een RTS.
```

Illustratie 3

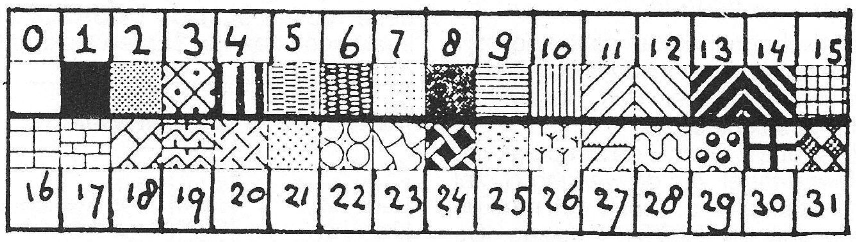
doet precies hetzelfde. ImprintRectangle zet een rechthoek in het achtergrondscherm, met RecoverRectangle haalt men die weer naar de voorgrond. InvertRectangle inverteert een rechthoek.

GRAPHICS

naam routine	adres	variabelen in	vernietigd	omschrijving
HorizontalLine	$c118	a -patroon, r11L-y coordinaat, r3 -x co. links, r4 -x co. rechts.	r5-r8,a,x,y, r11H.	Tekent een horizontale lijn.
VerticalLine	$c121	a -patroon, r3L -y boven, r3H-y co. onder, r4 -x coordinaat.	r4L-r8L, r11L,a,x,y.	Tekent een vertikale lijn.
SetPattern	$c139	a -nummer patroon. (0-15)	a	selecteer patroon
Rectangle	$c124	r2L-y co. boven, r2H -y co. onder r3 -x co. links, r5 -x co. rechts.	a,x,y,r5-r8, r11.	Tekent een opgevulde rechthoek met het geselecteerde patroon.
i_Rectangle	$c19f	.byte y co. bovenkant .byte y co. onderkant .word x co. linkerkant .word x co. rechterkant		
ImprintRectangle	$c250	zie Rectangle.		Zet rechthoek in achtergrondscherm
i_ImprintRect.	$c253	zie i_Rectangle.		
RecoverRectangle	$c12d	zie Rectangle.		Haal rechthoek van achter naar voor.
i_RecoverRect.	$c1A5	zie i_Rectangle.		
InvertRectangle	$c12a	zie Rectangle		Invert rechthoek.

Illustratie 5

Ik zal er eens een voordoen: We kijken in het schema onder 'variabelen in' bij 'HorizontalLine'. Hier staat dat in R11L, dus in de low-byte van R11 (d.i. 11*2+2 = 24 = hex $0018) de y-coördinaat moet. In R3 staat de x-coördinaat links (2 bytes), en in r4 staat de xcoordinaat rechts (ook 2 bytes). Nu maak ik een horizontale lijn van (x,y) (10,100) naar (260,100). Het patroon dat daarbij gebruikt wordt is gewoon massief zwart, dit heeft de code #$01. (zie illustratie 6).



Illustratie 6

```
lda #$64 {100 in $0018
sta $18 {
lda #$0a {10 in $0008
sta $08 {
lda #$00 {0*256 in $0009
sta $09 {10+0*256=10
lda #$04 {4 in $000a
sta $0a {
lda #$01 {1*256 in $000b
sta $0b {4+1*256=260
lda #$01 {patroon 1
jsr $c118
```

In GeoAssembly wordt hetzelfde zo genoteerd (dit demonstreert duidelijk het gemak van GeoAssembly):

```
loadB R11L,#100
loadW R3,#10
loadW R4,#260
lda  #$01
jsr  HorizontalLine
```

Inline Call

Een heel handige vondst van GEOS is de 'inline call'. Na het aanroepen van zo'n routine kijkt GEOS in de stack na vanaf welk adres er gesprongen is. Direct achter de routine van een inline call (deze zijn te herkennen aan het voorvoegsel 'i_') staan namelijk de variabelen. Dit kan vooral voor de programmeur zonder GeoAssembler veel tijd en moeite schelen. Niet van alle routines bestaan inline call versies, ik heb er 3 genoemd: i_Rectangle, i_ImprintRectangle en i_RecoverRectangle. Nu een voorbeeldje met i_Rectangle. Ik maak een opgevulde rechthoek van (x1,y1,x2,y2) (1,30,300,199). Dat ziet er zo uit:

```
jsr $c19f
$01,$00,
$1e,
$2c,$01,
$c7
```

in GeoAssembly:

```
jsr i_Rectangle
.byte #30
.byte #199
.word #1
.word #300
```

Testapplicatie

De testapplicatie is een eenvoudig programmaatje waarin de nu besproken technieken gebruikt worden. Het maakt een horizontaal hoofdmenu met 2 keuzes: 'Menu'tje' en 'Uit'.
Als 'Menu'tje' gekozen wordt krijgt men een vertikaal menu met 3 keuzes:

'Keuze 1' - Het scherm wordt wit gemaakt.
'Keuze 2' - Het scherm wordt zwart gemaakt.
'Normaal' - Het scherm krijgt weer het normale patroon: gestippeld.
Als 'uit' gekozen wordt, gaat men weer naar de DESKTOP. Wanneer U het allemaal nog niet helemaal begrijpt, kan het erg nuttig zijn de listing van de TESTAPP goed door te lezen. Precies naast elkaar staan de GeoAssembly en de normale machinetaal. De verklarende tekst wijst U voor de rest de weg.

Aan de slag

Wat nu te doen? Voor degenen die geen GeoAssembler bezitten is het deze keer zelfs gemakkelijker. Zij kunnen de kolom in het midden in een gewone monitor invoeren. Als dat gedaan is moet het programma 'GEOS-PROGRAM-MAKER', dat in de vorige aflevering gelist werd, gerund worden. Dit programma vraagt naar de naam, datum, dag etc. Dit kunt U allemaal naar eigen goeddunken bepalen. Het enige echt belangrijke is het invullen van de adressen:
beginadres NU? 4092
eindadres NU? 4224
laadadres? 4092
startadres? 4092
De PROGRAM-MAKER zet het programma dan op disk en het kan dan in GEOS (versie 1.2 of hoger) opgestart worden.

De weinigen die een GeoAssembler bezitten, moeten de linkerkolom van de listing intikken. Laat het adres na '.psect' weg, anders geeft de GeoAssembler een foutmelding. Dit adres moet u later, in de Geolinker invullen. Tenslotte moet er nog voor een fileheader gezorgd worden. Hiervoor kunt u heel goed de DEMO-header (Samseq.hdr), die bijgeleverd wordt gebruiken. Let dan wel op de juiste start- en laadadressen!

Volgende keer ga ik het hebben over de moeilijker diskroutines, het gebruik van dialog Boxes en tekststrings in GEOS.

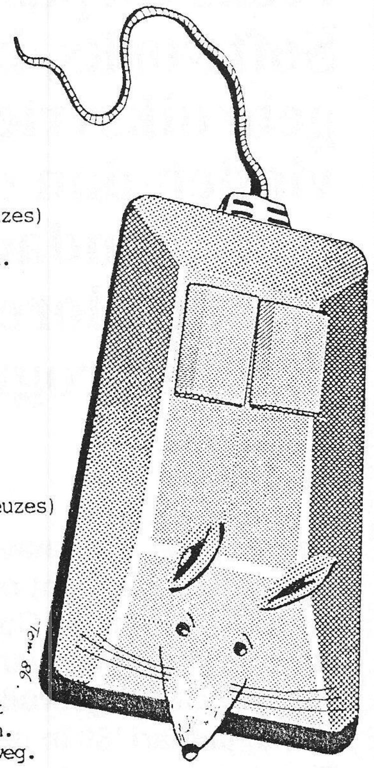
Bovendien volgt dan de listing van Quicktop, de eerste volwaardige applicatie uit deze cursus. De Quicktop is een summiere Public Domain Desktop, die de doorstroom naar applicaties zoals GeoWrite en GeoPaint aanzienlijk versnelt.

Maar probeert u nu de listing van deze keer eens uit!

Peter Boncz

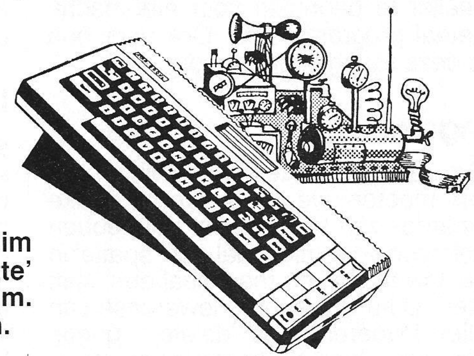
LISTING TESTAPP

geoassembly	normale assembly	uitleg
.psect 9ffc		
initroutine:		
loadB Displaybufferon,	9ffc lda #$c0	Bit 7 + Bit 6 aan in $002f,
#$c0	9ffd sta $2f	(Fore- + Background on) dus $c0
jsr NewDisk	1000 jsr $c1e1	Zet de diskdrive stop.
jsr MouseUp	1003 jsr $c18a	Doe de muis aan.
lda #$02	1006 lda #$02	Selecteer patroon 2
label1:		(gestippeld).
jsr SetPattern	1008 jsr $c139	
jsr i_Rectangle	100b jsr $c19f	Vul daar het hele
.byte #0	100e #$00	scherm mee.
.byte #199	100f #$c7	
.word #0	1010 #$00,#$00	
.word #319	1012 #$3f,#$01	
loadW R0,menutable	1014 lda #$22	$22
	1016 sta $02	en
	1018 lda #$10	$10 in R0 wijzen naar
	101a sta $03	de menutable in $1022.
lda #$00	101c lda #$00	Muis op het eerste menu.
jsr DoMenu	101e jsr $c151	Zet het menu op.
rts	1021 rts	einde initialisatie- routine.
menutable:		
.byte #0	1022 #$00	Top menu op Y=0.
.byte #14	1023 #$0e	Onderkant menu op Y=14.
.word #0	1024 #$00,#$00	Linkerkant menu op X=0.
.word #96	1026 #$60,#$00	Rechterkant bij X=96.
.byte #$02	1028 #$02	$02=$00 (hor.) OR $02 (2 keuzes)
.word menutjeTxt	1029 #$5a,#$10	Tekst "menutje" op $105a.
.byte #$80	102b #$80	$80 betekent: type submenu.
.word menutjeMenu	102c #$34,#$10	Submenu-table in $1034.
.word uitTxt	102e #$62,#$10	Tekst "Uit" in $1062
.byte #$00	1030 #$00	$00 betekent: type AKTIE.
.word uitRoutine	1031 #$4a,#$10	Uitroutine op $104a
.byte #00	1033 #$00	einde Menu-table
menutjeMenu:		
.byte #14	1034 #$0e	Top Menu op Y=14.
.byte #57	1035 #$39	Onderkant menu op Y=57.
.word #0	1036 #$00,#$00	Begin links op X=0.
.word #60	1038 #$3c,#$00	Einde Rechts op X=60.
.byte #$83	103a #$83	$83=$80 (vert.) OR $03 (3 keuzes)
.word keuze1Txt	103b #$66,#$10	Tekst "Keuze 1" in $1066.
.byte #$00	103d #$00	$00 betekent: type AKTIE.
.word keuze1Routine	103e #$50,#$10	Keuze1routine op $1050.
.word keuze2Txt	1040 #$6e,#$10	Tekst "Keuze 2" in $106e.
.byte #$00	1042 #$00	$00 betekent: type AKTIE.
.word keuze2Routine	1043 #$55,#$10	Keuze2routine in $1055.
.word normaalTxt	1045 #$76,#$10	Tekst "Normaal" in $1076.
.byte #$00	1047 #$00	$00 betekent: type AKTIE.
.word initroutine	1048 #$fc,#$0f	Initroutine ($0ffc) voor het normale patroon.
uitRoutine:		
jsr GotofirstMenu	104a jsr $c1bd	Doe het Pull-Downmenu weer weg.
jmp EnterDesktop	104d jsr $c22c	Ga DESKTOP laden.
keuze1Routine:		
lda #$01	1050 lda #$01	Kies patroon 1 (zwart).
jmp label1	1052 jmp $1008	(in $1008 is jsr SelectPattern)
keuze2Routine:		
lda #$07	1055 lda #$07	Kies patroon 7 (motief).
jmp label1	1057 jmp $1008	(in $1008 is jsr SelectPattern)
menutjeTxt:	105a "MENUTJE"+#$00	Alle hoofdletter zijn kleine
"menutje",0	in HEX: #$6d,#$65,#$6e,#$75	letters en alle kleine letters
	#$74,#$6a,#$65,#$00	zijn hoofdletters in GEOS.
uitTxt:	1062 "uIT"+#$00	De hexadecimale waarden
"Uit",0	in HEX: #$55,#$69,#$74,#$00	van de teksten staan alvast
keuze1Txt:	1066 "kEUZE 1"+#$00	eronder klaar.
"Keuze 1",0	in HEX: #$4b,#$65,#$75,#$7a,#$65,#$20,#$31,#$00	
keuze2Txt:	106e "kEUZE 2"+#$00	
"Keuze 2",0	in HEX: #$4b,#$65,#$75,#$7a,#$65,#$20,#$31,#$00	
normaalTxt:	1076 "nORMAAL"+#$00	
"normaal",0	in HEX: #$4e,#$6f,#$72,#$6d,#$61,#$61,#$6c,#$00	



Hoe vaak denkt u niet:"Wat gaat het toch weer lekker langzaam die basic." Ook de gecompileerde basic, van de snelste compiler, is vaak slomer dan echte machinetaal. Daarom hebben wij zelf wat routines gemaakt die ons vaak snel terzijde staan. Dit keer hebben wij een schermroutine en een invoer-routine geprogrammeerd. Dus dit keer weinig tekst en veel programma's.

TURBO routines



Wie heeft er niet regelmatig gedacht: "Waarom is er niet iemand zo slim om een invoer-routine te schrijven, zodat ik niet telkens met die 'rotte' basic input moet gaan zitten stoeien." Blijkbaar was er nog niemand zo slim. Hieronder hebben wij geprobeerd een dergelijke routine voor u te maken.

Intypen

Ook hier weer goed uit de ooghoeken kijken, en geen fouten maken. Elke fout kan funest zijn bij het opstarten. Daarom EERST de file wegschrijven en daarna opstarten. En denkt u nu niet:"Oh mij gebeurd toch niets." want dan kunt u nog raar op uw neus kijken, wanneer de computer 'hangt'. Wat dacht u van een stroomstoring? Heeft u die wel eens meegemaakt, en wanneer dan vele uren later blijkt dat ALLES voor nop is geweest. Dus ook tijdens het intypen even 'snel' de ingetikte data's wegschrijven. Hoeft u tenminste niet nog een keer alles in te tikken.

Wat is het

Wat is het en wat kan ik ermee zult u nu misschien denken. Ja het zijn drie programma's die elk weer wat anders kunnen doen. Zo is er een 'invoer-routine' die elke programmeur uit de 'invoer misere' haalt. Verder hebben we dan de VDC ramwriter, die circa 800% sneller data op uw scherm kan plaatsen. En dat is toch niet te zuinig. Als derde programma hebben we dit keer een soort subroutine die voor de VDC ramwriter de data in een buffer zet, wegschrijft en converteerd naar het juiste formaat, dit is de zogenaamde scherm editor.

80 CLS modus

Bijna alle dit keer door ons gepubliceerde programmatuur werkt uitsluitend in de 80 cls mode. Dat wil zeggen daar zijn 'ze' in ontworpen. En neem nu bijvoorbeeld de VDC ramwriter, die schrijft in het VDC videoram. Natuurlijk werkt het dan voor geen meter op het 40 cls. Zo ook de schermeditor, die werkt ook alleen in de 80 cls modus. Ook al weer zo simpel, want die schrijft en leest in en uit het VDC videoram. Wat moet u immers op 40 cls als u het onderste uit de kan wilt halen. De '128 kan' dan wel te verstaan. Dus wilt u sneller en een nettere invoer, dan zult u de invoer-routine moeten intikken. En wanneer u zelf schermen voor uw programma's wilt ontwikkelen kunt u de editor en de ramwriter even intikken.

40 CLS modus

Zoals wij u hierboven al op attendeerden werkt bijna alle programmatuur uitsluitend in de 80 cls mode. Maar de invoer routine werkt ook in de 40 cls modus! Dus heeft u (nog) geen 80 cls monitor dan kunt u 'voorlopig' op het 40 cls met de invoer routine aan het stoeien.

De Invoer-routine

De invoer routine is een hele speciale. U kunt de invoer routine bij basic programma's gebruiken. De dataloader zet de code, de machinetaal, vanaf adres $00b00 klaar voor gebruik neer. Na het aanroepen van de invoer routine heeft u, op de door u ingevoerde plaats, een aantal punjes staan, waarop u strak data kunt gaan invoeren. Het aanroepen gaat als volgt:

sys 2816,regel,kolom,lengte

En natuurlijk zult u op de plaats van 'regel' een waarde van 1 tot 25 moeten opgeven. Op de plaats van kolom moet u dan een waarde tussen 1 en 77 opgeven. Maar gebruikt u de invoer routine op het 40 cls dan geldt bij kolom alleen 1 tot 37! Als laatste zult u de lengte van de invoer moeten opgeven. Probeer het maar eens uit zullen we maar weer zeggen.

Waar blijven de data

Nu heeft u een invoer routine, maar waar blijft die ingevoerde data? Wij zetten die even voor u in een buffer. Met een klein stukje basic haalt u het er dan zo weer uit. Om de ingevoerde data vervolgens op het scherm af te drukken, of om in een string op te slaan. Deze invoer buffer staat trouwens vanaf $00c01. Om de data in een string te kunnen opslaan kunt u het beste het onderstaande stukje programma gebruiken. U kunt dat opnemen in uw eigen listing, maar wel de regelnummers aanpassen! En haal de data er wel uit voordat u de invoer routine nogmaals aanroept, anders bent u de eerder ingevoerde data kwijt!

```
10 in$=""
20 sys 2816, regel, kolom,
     lengte:mk=peek(253)
30 : for tl=1 to mk
40 :  in$=in$+chr$(peek(3072+tl))
50 : next tl
```

Wanneer u dus het bovenstaande programmagedeelte gebruikt gaat alles goed. En zal uw data in de in$ zijn opgeslagen. U kunt nu vanuit uw basic programma met deze string verder werken. Ook voor het bovenstaande programma geldt: geldige waardes invoeren a.u.b. anders werkt het niet! Dus geen regel en kolom laten staan maar daar een waarde invoeren. En denk daarbij om de regels die wij al eerder in deze tekst hebben opgeven.

Fast listing

Om het wat gemakelijker te maken hebben wij een zogenaamde fast listing gepubliceerd. In deze gedocumenteerde listing staat een heleboel tekst die voor de machinetaal programmeur zeer handig kan zijn, maar

dan moet hij of zij er wel wat mee gaan doen natuurlijk! De machinetaal programmeur heeft immers veel meer aan zo'n 'Fast listing' dan aan een basic loader. En daarom dus ook deze listing. Een 'Fast listing' is ook veel sneller te begrijpen voor niet-machinetaal programmeurs. Ook voor hun is deze listing super handig.

Ingave

Om een karakter te kunnen binnenhalen moeten we eerst weten welke waardes zijn toegestaan. Wij hebben gekozen voor minimaal een 'spatie' in de 'lowercase' en maximaal een 'shift ster'. U kunt dus in de lowercase van alles invoeren. Een 'delete' 'quote' 'esc' en 'return' zijn kontrole karakters. Wanneer u dus een dergelijk karakter invoert wordt deze er weer uitgefilterd, of die doet een aparte opdracht.

RETURN

Zo is er de code $0d (13) voor return, met deze return komt u weer uit de invoer routine. Heeft u geen data ingevoerd, dan staat er ook niets in de buffer. Dit is dus de 'normale' manier van ontsnapping aan onze routine. Ook run/stop/restore werkt natuurlijk, maar dan bent u alles weer kwijt.

ESC

Wanneer u data heeft ingevoerd en er niet tevreden mee bent, of u heeft een fout gemaakt, drukt u gewoon op de 'ESC' toets. Hierdoor zullen de puntjes weer opnieuw op het scherm worden gezet, en u kunt weer opnieuw gaan invoeren. Een soort van terug naar af dus!

DELETE

Met 'delete' kunt u een karakter terug. U kunt dus bij een gemaakte typefout direct ingrijpen. Gaat het helemaal fout pakt u gewoon de ESC toets, want dan is alles weer weg! Esc is dus een soort hersteller.

QUOTE

Wanneer u 'shift 2' intoetst komt u normaal in de 'quote mode', maar dat is niet de bedoeling. Het 'quote' teken is dus taboe voor de invoer routine! Dus geen vreemde tekens meer in de invoer modus.

De editor

De schermeditor is ook al zo'n handig programma. U kunt met behulp van deze scherm editor, die alleen in de 80 cls mode werkt, schermen gaan maken. Een tool die bij ons zijn sporen al heeft verdient. Geloof ons het heeft ons al een keer bijna twee weken werk bespaard. Het zal je toch maar gebeuren dat iemand je vraagt om een programma aan te passen en te vertalen. Zit je mooi hoog en droog dan kan je er niets gebeuren. Maar wat zonder een editor? Geen idee dan maar snel een editor ontwikkelen. En zo ontstond bij ons deze REUS!! Gewoon doen en niet zeuren werd er ons gezegd. En het heeft ons toch nog wat opgeleverd, en u ook nietwaar dan.

Ietsje sneller

Wanneer het snel kan dan moet het snel en wanneer het nog sneller kan dan moet het sneller! Met deze wetenschap zijn wij aan de slag gegaan. En wij mogen zeggen niet voor niets is de routine zo snel als hij nu is. Wij moesten er even voor gaan zitten maar het is zo ver.

HET SCHERM Het monitor scherm heeft na het opstarten van de editor 30 regels. U kunt die niet gebruiken maar daar staat wat informatie omtrent de ingevoerede tekens. Zo staat er op welke regel en kolom u aan het werk bent. Ook wordt elk teken in twee groepen verdeeld. Zo is er dan een ascii en een schermcode weergave op het scherm te zien. U zult dus onder in uw scherm, op de 5 extra regels, deze code's zien staan. Deze code's zijn ook weer in twee groepen verdeeld, namelijk als 'HEX' en als 'DEC' waarde. Wanneer u dus veel met de C64 heeft zitten werken zult u de kreet schermcode's nog wel kennen.

SAVEN

Om een door u zelf gemaakt scherm te kunnen wegschrijven dient u op de pijl naar links toets te drukken. U kunt deze toets vinden aan de linker kant van het toetsenboard. Vervolgens wordt uw scherm in een buffer gezet, en kunt u een filenaam invoeren. Daarna wordt de data uit de buffer op disk weggeschreven. U dient dan wel op de return toets te drukken, dit als afsluiting van de opdracht wegschrijven.

STOPPEN

Om met het programma te kunnen stoppen kunt u gewoon op run/stop/restore drukken. Maar er is een betere oplossing voor. Deze is het indrukken van 'shift return' Hierdoor start de C128 vanzelf weer, als nieuw, op en lkunt u met een ander programma aan de slag.

TIP

Lees de **ESC code's** uit het vorige nummer nog eens door. Die zijn nu helemaal handig.

Na het intypen van de data's heeft u er een geweldig editor bij. Deze editor vraagt er om, om gebruikt te worden. Dus doe dat dan ook. Nu moeten wij het u nog vertellen maar na er eenmaal mee gewerkt te hebben is dat beslist niet meer nodig.

VDC Ramwriter

Wanneer u eenmaal een scherm, helpmenu of ander soort scherm heeft gemaakt met behulp van onze schermeditor, wilt u natuurlijk kijken hoe het is geworden. En niet te vergeten hoe snel het op uw scherm wordt gezet. Zonder te overdrijven kunnen wij zeggen dat wij nog nooit zo snel een heel scherm hebben zien opvullen.

Onze routine is circa 800% sneller dan de kernal routine's van de CBM 128. En dat is ook niet gek, want die waren niet op een dergelijke manier van afprinten berekend.

Fast listing

Ook van de VDC ramwriter is een zogenaamde 'fast listing' afgedrukt. Weer zo handig voor de machinetaal programmeurs onder ons. En weer zo handig voor de basic en aspirant machinetaal programmeur.

De aanroep

U kunt de VDC ramwriter op de onderstaande manier aanroepen:

sys 2816,19,00,0

Wanneer u op de bovenstaande manier de routine aanroept zal de ramwriter uit bank 0 vanaf adres $1300 de data ophalen en op uw scherm plaatsen. 19 is dus de decimale waarde van $13. En aangenomen dat bij dit voorbeeld de data, die u met de editor heeft gemaakt, dan door u vanaf $01300 is ingeladen door middel van bijvoorbeeld:

bload"filenaam",b0,p4864

Door deze opdracht wordt de data dan vanaf adres $01300 ingeladen!

voorbeeld 2:

bload"filenaam",b0,p4865

en dan

sys 2816,19,1,0

zal er voor zorgen dat de ramwriter de data vanaf $1301 uit bank 0 ophaalt en op uw scherm plaatst.

Demo

Om het nog gemakkelijker te maken hebben wij voor u een demo in elkaar gezet die het kunst en vliegwerk van de editor/VDC ramwriter demonstreerd. U kunt daarvoor, na de ramwriter te hebben ingeladen, de basic data demo opstarten.

Veel plezier

JOHAN & JOHAN

Listings Turbo routines

Invoer-routine (basic listing)

```
1000 rem"*********************************"
1010 rem"**    Invoer routine voor de CBM
   128(d) in de 80 cls mode!     **"
1020 rem"**       door Johan & Johan voor
   COMMODORE INFO in 1988.       **"
1030 rem"*********************************"
1040 :
1050 : for x=2816 to 3014
1060 :  read a
1070 :   cs=cs+a
1080 :  poke x,a
1090 : next x
1100 :
1110 : if cs25705 then
   printchr$(7):list:end
1120 :
1130 data
   201,000,240,090,224,000,240,086,192,000,2
   40,082,032,173,011,166
1140 data
   250,164,251,032,190,011,169,030,032,045,1
   99,169,046,162,255,032
1150 data
   045,199,232,228,252,208,248,169,032,032,0
   45,199,166,250,164,251
1160 data
   032,190,011,032,045,199,032,239,238,240,2
   51,201,027,208,007,169
1170 data
   000,133,253,076,015,011,201,034,240,236,2
   01,013,208,017,165,253
1180 data
   201,000,208,010,169,000,157,000,012,169,0
   05,032,045,199,096,201
1190 data
   020,208,013,166,253,224,000,240,205,198,2
   53,166,253,076,157,011
1200 data
   024,201,032,144,193,024,201,096,144,010,0
   24,201,193,144,183,024
1210 data
   201,219,176,178,166,253,228,252,240,172,2
   32,134,253,157,000,012
1220 data
   072,169,005,032,045,199,104,032,045,199,0
   76,054,011,160,000,185
1230 data
   195,011,032,045,199,200,192,004,208,245,0
   76,054,011,202,134,250
1240 data
   136,132,251,133,252,024,032,106,204,169,0
   00,133,253,096,134,235
1250 data 132,236,096,030,157,046,157
1260 :
1270 bsave "invoer_128/m",b0,p2816 to p3014
```

EINDE basic listing invoer-routine

Invoer-routine (fast assembler)

```
1000              bnk 0          ;schrijf code in bank 0
1010              org $0b00      ;vanaf adres $0b00 (2816)
1020 ;
1030 buffer       equ $0c00      ;de hieronder gebruikte 'namen'
1040 bsout        equ $c72d      ;worden hier opgegeven
1050 chrin        equ $eeef      ;omdat dat gewoon 'lekker' werkt
1060 plot         equ $cc6a
1070 x_pos        equ $fa        ;met de x-positie wordt hier dus
1080 y_pos        equ $fb        ;adres $fa bedoeld enz enz...
1090 max_len      equ $fc        ;maximal in te geven tekens
1100 kar_teller   equ $fd        ;de karakter teller
1110 lo_min       equ 32         ;dit zijn de in te typen tekens
1120 lo_max       equ 96         ;in lower- en upper case
1130 hi_min       equ 193        ;'lo' voor lowercase en natuurlijk
1140 hi_max       equ 219        ;'up' voor uppercase
1150 ;
1160 a_x_y        cmp #0         ;aantal karakters gelijk aan 0
1170              beq go_out     ;dan illegale opstart en exit!
1180              cpx #0         ;x-positie gelijk aan 0
1190              beq go_out     ;dan illegale opstart en exit!
1200              cpy #0         ;y-positie gelijk aan 0
1210              beq go_out     ;dan illegale opstart en exit!
1220 opstart      jsr init       ;de init routine zet start waarden
1230 start        ldx x_pos      ;x-register met x-positie laden
1240              ldy y_pos      ;y-register met y-positie laden
1250              jsr set_pos    ;en vervolgens activeren
1260              lda #30        ;accu met kleur laden
1270              jsr bsout      ;en activeren
1280              lda #46        ;asc waarde voor punt
1290              ldx #255       ;x-reg. (teller) met nul laden
1300 set_punt     jsr bsout      ;print een punt
1310              inx            ;ophogen van het x-register
1320              cpx max_len    ;maximaal aantal punten
1330              bne set_punt   ;nog niet, dan volgende punt afprinten
1340              lda #32        ;asc waarde voor spatie
1350              jsr bsout      ;en afprinten
```

```
1360                    ldx x_pos          ;x-reg. laden met de regel
1370                    ldy y_pos          ;en het y-reg. laden met de kolom
1380                    jsr set_pos        ;en naar de routine die deze aanpast
1390                    jsr bsout          ;druk karakter op het scherm
1400 main               jsr chrin          ;haal een teken via het toetsenboard
1410                    beq main           ;geen teken dan naar main
1420                    cmp #27            ;kontroleer op een 'esc' teken
1430                    bne cont           ;zo niet dan naar volgende kontrole
1440                    lda #0             ;wanneer het een esc teken is accu op 0
1450                    sta kar_teller     ;en karakter teller op nul zetten
1460                    jmp start          ;weer naar start, (puntjes zetten)
1470 cont               cmp #34            ;kontroleer op een 'quote'
1480                    beq main           ;zo ja dan teken anuleren en naar main
1490                    cmp #13            ;kontroleer op een 'return' teken
1500                    bne check_back     ;geen return dan volgende kontrole
1510                    lda kar_teller     ;accu met karakter teller laden
1520                    cmp #0             ;kontroleer of aantal tekens nul is
1530                    bne go_out         ;geen nul dan naar basic
1540                    lda #0             ;en anders buffer afsluiten
1550                    sta buffer,x       ;op positie van de waarde van het x-reg
1560                    lda #5             ;kleur weer op de normale waarde
1570                    jsr bsout          ;terug zetten d.m.v bsout op het scherm
1580 go_out             rts                ;nu weer terug naar aanroep (basic)
1590 ;
1600 check_back         cmp #20            ;kontroleer op een 'delete' teken
1610                    bne check_ascii    ;geen delete dan verder met ascii kontrole
1620                    ldx kar_teller     ;x-reg. met karakter teller laden
1630                    cpx #0             ;is de teller 0 ?
1640                    beq main           ;zo ja anuleer ingave
1650                    dec kar_teller     ;en zo niet, teller met 1 aflagen
1660                    ldx kar_teller     ;x-reg. met teller laden
1670                    jmp delete         ;en 1 karakter van het scherm afvegen.
1680 ;
1690 check_ascii        clc                ;carry wissen
1700                    cmp #lo_min        ;kontroleer minimale waarde lowercase
1710                    bcc main           ;kleiner, dan fout
1720                    clc                ;carry wissen
1730                    cmp #lo_max        ;kontroleer maximale waarde lowercase
1740                    bcc out            ;kleiner, dan oke!
1750                    clc                ;carry wissen
1760                    cmp #hi_min        ;kontroleer minimale waarde uppercase
1770                    bcc main           ;kleiner, dan fout!
1780                    clc                ;carry wissen
1790                    cmp #hi_max        ;kontroleer maximale waarde uppercase
1800                    bcs main           ;groter dan fout!
1810 out                ldx kar_teller     ;x-reg. met aantal karakters laden
1820                    cpx max_len        ;maximale ingave?
1830                    beq main           ;zo ja dan weer terug naar af!
1840                    inx                ;buffer positie ophogen
1850                    stx kar_teller     ;x-reg. op karakter teller zetten
1860                    sta buffer,x       ;en het karakter in de buffer schrijven
1870                    pha                ;ons karakter even opbergen
1880                    lda #5             ;accu met de code voor wit laden
1890                    jsr bsout          ;en kleur via bsout zetten
1900                    pla                ;nu ons karakter weer terug halen
1910                    jsr bsout          ;en op het scherm plaatsen
1920                    jmp main           ;op naar de volgende ingave
1930 ;
1940 delete             ldy #4             ;sub-routine 'delete'
1950 dummy0             lda tabel,y        ;tabel waarde inladen
1960                    jsr bsout          ;afprinten op het scherm
1970                    dey                ;y-register aflagen
1980                    bpl dummy0         ;nog geen nul, dan volgende
1990                    jmp main           ;en maar verder
2000 ;
2010 init               dex                ;x-positie bijstellen
2020                    stx x_pos          ;op onze x-positie wegzetten
2030                    dey                ;y-positie bijstellen
2040                    sty y_pos          ;op onze y-positie wegzetten
2050                    sta max_len        ;maximaal aantal karakters opbergen
2060                    clc                ;carry wissen, voor
```

```
2070                jsr plot           ;cursor positie goed zetten
2080                lda #0             ;accu met waarde nul laden, en
2090                sta kar_teller     ;op de de karakterteller zetten
2100                rts                ;terug uit sub-routine
2110 ;
2120 set_pos        stx $eb            ;regel positie wegzetten.
2130                sty $ec            ;kolom positie wegzetten.
2140                rts                ;terug uit sub-routine.
2150 ;
2160 tabel          dat $9d2e9d1e
2170                end                ;einde voor de 'fast' assembler
```

Ramwriter (basic listing)

```
1000 rem"**********************************"
1010 rem"**     VDC ramwriter voor de CBM
   128(d) in de 80 cls mode!    **"
1020 rem"**       door Johan & Johan voor
   COMMODORE INFO in 1988.      **"
1030 rem"**********************************"
1040 :
1050 : for x=2816 to 2911
1060 :  read a
1070 :   cs=cs+a
1080 :  poke x,a
1090 : next x
1100 :
1110 if cs9594 then printchr$(7):list:end
1120 :
1130 data
   133,253,134,252,132,254,160,000,140,000,2
   55,032,086,011,162,018
1140 data
   032,048,011,032,060,011,032,086,011,162,0
   19,032,048,011,032,060
1150 data
   011,032,086,011,201,255,240,032,032,046,0
   11,076,030,011,162,031
1160 data
   142,000,214,044,000,214,016,251,141,001,2
   14,096,024,165,252,105
1170 data
   001,133,252,144,002,230,253,096,032,060,0
   11,032,086,011,201,255
1180 data
   240,003,076,014,011,096,169,252,166,254,1
   60,000,032,116,255,096
1190 :
```

```
1000                bnk 0              ;schrijf code in bank 0
1010                org $0b00          ;vanaf adres $0b00 (2816)
1020 init           sta $fd            ;adres in het a en x-register
1030                stx $fc            ;wegschrijven als basis adres
1040                sty $fe            ;banknummer uit y-reg. wegzetten
1050                ldy #0             ;y-reg. laden met waarde 0
1060                sty $ff00          ;en alle rom inschakelen
1070 ;
1080 set_ram        jsr load_ram       ;data uit buffer ophalen
1090 set_ram2       ldx #18            ;x-register laden met waarde 18
1100                jsr set_reg        ;en wegzetten
1110                jsr inc_count      ;bufferteller met 1 ophogen
1120                jsr load_ram       ;data uit buffer ophalen
1130                ldx #19            ;x-register laden met waarde 19
1140                jsr set_reg        ;en wegzetten
1150 ;
1160 get_data       jsr inc_count      ;bufferteller met 1 ophogen
1170                jsr load_ram       ;data uit ram ophalen
1180                cmp #255           ;gelijk aan 255 ?
1190                beq check_next     ;ja dan sprong, en anders
1200                jsr write_data     ;data in videoram zetten
1210                jmp get_data       ;en volgende data
1220 ;
1230 write_data     ldx #31            ;x-register laden met waarde 31
1240 set_reg        stx $d600          ;en plaatsen op adres $d600
1250 wait           bit $d600          ;test status van 8563 chip
1260                bpl wait           ;klaar, anders nog even wachten!
1270                sta $d601          ;schrijf data in videoram
1280                rts                ;en terug
1290 ;
1300 inc_count      clc                ;wis de carry vlag en
1310                lda $fc            ;load accu met waarde van adres $fc
1320                adc #1             ;deze waarde met 1 ophogen
1330                sta $fc            ;en weer terug zetten op $fc
1340                bcc next           ;niet groter dan 255, dan terug
1350                inc $fd            ;en anders high-byte 1 ophogen
1360 next           rts                ;daarna terug
1370 ;
1380 check_next     jsr inc_count      ;bufferteller met 1 ophogen
```

```
1390                 jsr load_ram        ;en data uit buffer ophalen
1400                 cmp #255            ;vergelijk waarde met 255
1410                 beq exit            ;gelijk, dan terug naar aanroep
1420                 jmp set_ram2        ;data wegschrijven in videoram
1430 exit            rts
1440 ;
1450 load_ram        lda #$fc            ;pointer naar het load adres
1460                 ldx $fe             ;data ophalen uit bank ($fd)
1470                 ldy #0              ;index op nul
1480                 jsr $ff74           ;indfet routine van de kernal
1490                 rts                 ;aanroepen en daarna terug
1500                 end                 ;einde van de 'fast' assembler
```

Listing Editor

```
1000 rem"********************************"
1010 rem"**     Scherm editor voor de CBM
    128(d) in de 80 cls mode!    **"
1020 rem"**       door Johan & Johan in
    1988 voor COMMODORE INFO!      **"
1030 rem"********************************"
1040 :
1050 : for x=4864 to 6709
1060 :  read a
1070 :   cs=cs+a
1080 :  poke x,a
1090 : next x
1100 :
1110 if cs175430 then
    printchr$(7):list:end
1120 :
1130 data
    007,208,112,064,064,064,064,064,064,064,1
    14,064,064,064,064,064
1140 data
    064,064,114,064,064,064,064,064,064,064,1
    14,064,064,064,064,064
1150 data
    064,064,114,064,064,064,064,064,064,064,1
    14,064,064,064,064,064
1160 data
    064,064,114,064,064,064,064,064,064,064,1
    14,064,064,064,064,064
1170 data
    064,064,114,064,064,064,064,064,064,064,0
    64,064,064,064,064,064
1180 data
    064,110,255,008,032,074,032,018,005,007,0
    05,012,032,032,032,011
1190 data
    015,012,015,013,032,032,032,001,019,003,0
    09,009,032,032,032,032
1200 data
    008,005,024,032,032,032,032,032,004,005,0
    03,032,032,032,032,002
1210 data
    005,005,012,004,032,032,032,032,008,005,0
    24,032,032,032,032,032
1220 data
    004,005,003,032,032,032,032,032,032,032,0
    32,020,009,010,004,032
1230 data
    032,032,032,032,074,255,008,112,038,032,0
    32,032,032,032,032,032
1240 data
    032,032,032,032,032,032,032,032,032,032,0
    32,032,032,032,032,032
1250 data
    032,032,032,032,032,032,032,032,032,032,0
    32,032,032,032,032,032
1260 data
    032,032,032,032,032,032,032,032,032,032,0
    32,032,032,032,032,032
1270 data
    032,032,032,032,032,032,032,032,032,032,0
    32,032,032,032,032,032
1280 data
    032,032,032,032,032,032,032,038,255,008,1
    92,074,032,032,048,048
1290 data
    048,032,032,032,032,032,048,048,048,032,0
    32,032,032,032,032,045
1300 data
    062,032,032,032,032,032,032,048,048,032,0
    32,032,032,032,048,048
1310 data
    048,032,032,032,032,032,032,045,062,032,0
    32,032,032,032,032,048
1320 data
    048,032,032,032,032,032,048,048,048,032,0
    32,032,032,032,032,032
1330 data
    032,032,032,032,032,032,032,032,032,032,0
    74,255,009,016,109,064
1340 data
    064,064,064,064,064,064,113,064,064,064,0
    64,064,064,064,113,064
1350 data
    064,064,064,064,064,064,113,064,064,064,0
    64,064,064,064,113,064
1360 data
    064,064,064,064,064,064,113,064,064,064,0
    64,064,064,064,113,064
1370 data
    064,064,064,064,064,064,113,064,064,064,0
    64,064,064,064,113,064
1380 data
    064,064,064,064,064,064,064,064,064,064,0
    64,064,064,125,255,255
1390 data
    032,032,032,032,032,032,032,032,032,032,0
    32,032,032,032,032,032
1400 data
    032,032,032,032,032,032,032,032,032,032,0
    32,032,032,032,032,032
1410 data
    032,033,034,035,036,037,038,039,040,041,0
    42,043,044,045,046,047
1420 data
    048,049,050,051,052,053,054,055,056,057,0
    58,059,060,061,062,063
1430 data
    000,001,002,003,004,005,006,007,008,009,0
    10,011,012,013,014,015
1440 data
    016,017,018,019,020,021,022,023,024,025,0
    26,027,028,029,030,031
```

```
1450 data
     064,065,066,067,068,069,070,071,072,073,0
     74,075,076,077,078,079
1460 data
     080,081,082,083,084,085,086,087,088,089,0
     90,091,092,093,094,095
1470 data
     032,032,032,032,032,032,032,032,032,032,0
     32,032,032,032,032,032
1480 data
     032,032,032,032,032,032,032,032,032,032,0
     32,032,032,032,032,032
1490 data
     096,097,098,099,100,101,102,103,104,105,1
     06,107,108,109,110,111
1500 data
     112,113,114,115,116,117,118,119,120,121,1
     22,123,124,125,126,127
1510 data
     064,065,066,067,068,069,070,071,072,073,0
     74,075,076,077,078,079
1520 data
     080,081,082,083,084,085,086,087,088,089,0
     90,091,092,093,094,095
1530 data
     096,097,098,099,100,101,102,103,104,105,1
     06,107,108,109,110,111
1540 data
     112,113,114,115,116,117,118,119,120,121,1
     22,123,124,125,126,094
1550 data
     004,016,080,012,005,001,019,005,032,005,0
     14,020,005,018,032,006
1560 data
     009,012,005,014,001,013,005,058,255,255,1
     69,000,141,000,255,169
1570 data
     128,133,247,169,253,141,048,208,169,147,0
     32,045,199,169,014,032
1580 data
     045,199,032,033,203,032,229,202,162,020,1
     69,010,141,047,010,032
1590 data
     205,023,162,021,169,000,032,205,023,162,0
     06,169,030,032,205,023
1600 data
     162,007,169,035,032,205,023,169,010,133,2
     50,162,018,165,250,032
1610 data
     205,023,162,019,169,000,032,205,023,162,0
     31,169,143,032,205,023
1620 data
     162,030,169,255,032,205,023,230,250,165,2
     50,201,020,208,220,169
1630 data
     000,162,019,032,067,024,169,033,133,082,1
     69,026,133,083,169,082
1640 data
     162,001,160,001,032,101,255,032,217,023,0
     32,235,238,240,248,133
1650 data
     027,032,142,023,032,117,023,133,251,134,2
     52,132,253,160,227,032
1660 data
     175,023,165,027,032,142,023,032,155,023,1
     33,251,134,252,169,032
1670 data
     133,253,160,219,032,175,023,166,027,224,0
     32,208,004,138,076,127
1680 data
     022,189,160,020,201,032,208,007,169,045,1
     70,168,076,130,022,032
1690 data
     117,023,133,251,134,252,132,253,160,251,0
     32,175,023,166,027,224
1700 data
     032,208,004,138,076,164,022,189,160,020,2
     01,032,208,006,169,045
1710 data
     170,076,167,022,032,155,023,133,251,134,2
     52,160,032,132,253,160
1720 data
     243,032,175,023,165,027,201,141,240,036,2
     01,095,240,035,201,145
1730 data
     240,073,201,017,240,080,201,157,240,087,2
     01,029,240,094,032,045
1740 data
     199,165,236,032,055,023,165,235,032,086,0
     23,076,055,022,076,000
1750 data
     224,032,226,202,032,129,024,032,199,024,1
     69,147,032,045,199,169
1760 data
     160,162,021,032,067,024,032,064,025,169,2
     55,162,010,032,205,023
1770 data
     169,005,032,045,199,032,246,025,076,186,0
     21,032,045,199,165,235
1780 data
     032,086,023,076,055,022,032,045,199,165,2
     35,032,086,023,076,055
1790 data
     022,032,045,199,165,236,032,055,023,076,0
     55,022,032,045,199,165
1800 data
     236,032,055,023,076,055,022,032,251,249,1
     33,250,134,251,162,018
1810 data
     169,008,032,205,023,232,169,204,032,205,0
     23,165,251,032,203,023
1820 data
     165,250,032,203,023,096,032,251,249,133,2
     52,134,253,162,018,169
1830 data
     008,032,205,023,232,169,196,032,205,023,1
     65,253,032,203,023,165
1840 data
     252,032,203,023,096,160,048,056,233,100,1
     44,003,200,176,249,105
1850 data
     100,162,048,056,233,010,144,003,232,176,2
     49,105,058,096,024,201
1860 data
     192,144,007,201,223,176,003,056,233,096,0
     96,168,041,015,170,152
1870 data
     041,240,024,074,074,074,074,168,189,034,0
     26,190,034,026,096,162
1880 data
     018,169,008,032,205,023,232,152,032,205,0
     23,165,253,032,203,023
1890 data
     165,252,032,203,023,165,251,032,203,023,0
     96,162,031,142,000,214
1900 data
     044,000,214,016,251,141,001,214,096,160,0
     03,173,011,220,041,031
1910 data
     076,230,023,185,008,220,041,240,074,074,0
     74,074,072,185,008,220
```

```
1920 data
    041,015,072,136,208,237,173,008,220,104,1
    62,008,032,044,024,104
1930 data
    162,007,032,044,024,169,010,162,003,032,0
    44,024,169,010,162,006
1940 data
    032,044,024,104,162,005,032,044,024,104,1
    62,004,032,044,024,104
1950 data
    162,002,032,044,024,104,162,001,032,044,0
    24,096,024,105,048,072
1960 data
    138,162,019,105,003,032,205,023,169,009,2
    02,032,205,023,104,032
1970 data
    203,023,096,133,250,134,251,160,000,177,2
    50,162,018,032,205,023
1980 data
    232,032,104,024,177,250,032,205,023,032,1
    04,024,177,250,201,255
1990 data
    240,018,032,203,023,076,089,024,024,165,2
    50,105,001,133,250,144
2000 data
    002,230,251,096,032,104,024,177,250,201,2
    55,240,003,076,075,024
2010 data
    096,169,000,133,251,133,253,133,252,169,0
    32,133,254,162,018,165
2020 data
    252,032,205,023,232,165,251,032,205,023,1
    60,000,169,031,141,000
2030 data
    214,044,000,214,016,251,173,001,214,145,2
    53,230,253,208,002,230
2040 data
    254,230,251,208,002,230,252,165,252,201,0
    08,208,006,165,251,201
2050 data
    208,240,003,076,141,024,096,169,255,162,0
    31,133,251,134,252,169
2060 data
    255,162,047,133,158,134,159,162,000,160,0
    00,032,036,025,224,001
2070 data
    240,053,177,251,201,032,240,243,072,032,0
    57,025,165,252,056,233
2080 data
    032,145,158,032,057,025,165,251,145,158,1
    04,032,057,025,145,158
2090 data
    032,036,025,224,001,240,016,177,251,201,0
    95,208,238,009,160,032
2100 data
    057,025,145,158,076,219,024,169,255,032,0
    57,025,145,158,032,057
2110 data
    025,145,158,096,230,251,208,002,230,252,1
    65,252,201,039,208,008
2120 data
    165,251,201,208,208,002,162,001,096,230,1
    58,208,002,230,159,096
2130 data
    032,220,025,169,030,032,045,199,166,250,1
    64,251,032,241,025,169
2140 data
    046,162,016,032,045,199,202,224,255,208,2
    48,169,032,032,045,199
2150 data
    166,250,164,251,032,241,025,032,045,199,0
    32,235,238,240,251,201
2160 data
    027,208,007,169,000,133,253,076,067,025,2
    01,034,240,236,201,013
2170 data
    208,012,165,253,201,000,208,005,169,000,1
    57,224,015,096,201,020
2180 data
    208,013,166,253,224,000,240,210,198,253,1
    66,253,076,204,025,024
2190 data
    201,032,144,198,024,201,096,144,010,024,2
    01,193,144,188,024,201
2200 data
    219,176,183,166,253,224,016,240,177,232,1
    34,253,157,224,015,072
2210 data
    169,005,032,045,199,104,032,045,199,076,1
    06,025,160,000,185,050
2220 data
    026,032,045,199,200,192,004,208,245,076,1
    06,025,162,013,160,023
2230 data
    024,032,024,192,166,235,134,250,164,236,1
    32,251,169,000,133,253
2240 data
    096,134,235,132,236,096,165,253,162,225,1
    60,015,032,189,255,169
2250 data
    000,162,000,032,104,255,169,000,162,008,1
    60,000,032,186,255,169
2260 data
    000,162,048,133,250,134,251,169,250,166,1
    58,164,159,032,216,255
2270 data
    096,223,048,049,050,051,052,053,054,055,0
    56,057,065,066,067,068
2280 data 069,070,030,157,046,157
2290 :
2300 bsave "editor_128/m",b0,p4864 to
    p6709
2310 sys 5562
```

EINDE listing Editor

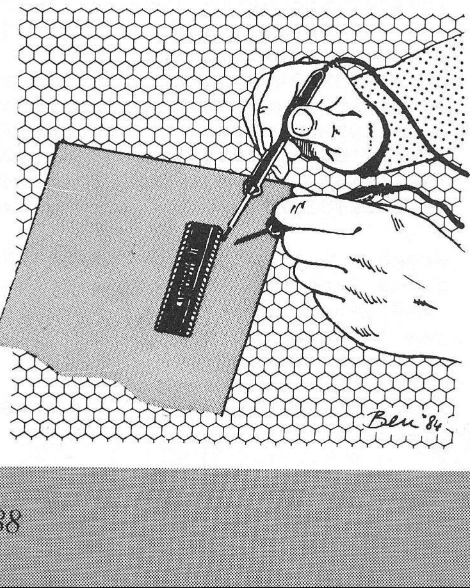
Demo Ramwriter

Listing demo Ramwriter

```
1000 rem"********************************"
1010 rem"**  VDC ramwriter demo voor de
   CBM 128(d) in de 80 cls mode!  **"
1020 rem"**         door Johan & Johan voor
   COMMODORE INFO in 1988.        **"
1030 rem"********************************"
1040 : for x=4864 to 6089
1050 :  read a
1060 :    cs=cs+a
1070 :  poke x,a
1080 : next x
1090 :
1100 if cs85805 then printchr$(7):list:end
1110 :
1120 printchr$(147)
1130 sys2816,19,0,0
1140 sleep 1
1150 sys2816,20,91,0
1160 sleep 1
1170 sys2816,20,162,0
1180 sleep 1
1190 sys2816,21,78,0
1200 sleep 1
1210 sys2816,21,149,0
1220 sleep 1
1230 sys2816,23,153,0
1240 sleep 1
1250 sys2816,23,185,0
1260 goto 1260
1270 data
    000,000,112,064,064,064,064,064,064,064,0
    64,064,064,064,064,064
1280 data
    064,064,064,064,064,064,064,064,064,064,0
    64,064,064,064,064,064
1290 data
    064,064,064,064,064,064,064,064,064,064,0
    64,064,064,064,064,064
1300 data
    064,064,064,064,064,064,064,064,064,064,0
    64,064,064,064,064,064
1310 data
    064,064,064,064,064,064,064,064,064,064,0
    64,064,064,064,064,110
1320 data
    255,000,080,093,255,000,157,093,255,000,1
    60,093,255,000,237,093
1330 data
    255,000,240,093,255,001,061,093,255,001,0
    64,093,255,001,141,093
1340 data
    255,001,144,093,255,001,221,093,255,001,2
    24,093,255,002,045,093
1350 data
    255,002,048,093,255,002,125,093,255,002,1
    28,093,255,002,205,093
1360 data
    255,002,208,093,255,003,029,093,255,003,0
    32,093,255,003,109,093
1370 data
    255,003,112,093,255,003,189,093,255,003,1
    92,093,255,004,013,093
1380 data
    255,004,016,093,255,004,093,093,255,004,0
    96,093,255,004,173,093
1390 data
    255,004,176,093,255,004,253,093,255,005,0
    00,093,255,005,077,093
1400 data
    255,005,080,093,255,005,157,093,255,005,1
    60,093,255,005,237,093
1410 data
    255,005,240,093,255,006,061,093,255,006,0
    64,093,255,006,141,093
1420 data
    255,006,144,093,255,006,221,093,255,006,2
    24,093,255,007,045,093
1430 data
    255,007,048,093,255,007,125,093,255,007,1
    28,109,064,064,064,064
1440 data
    064,064,064,064,064,064,064,064,064,064,0
    64,064,064,064,064,064
1450 data
    064,064,064,064,064,064,064,064,064,064,0
    64,064,064,064,064,064
1460 data
    064,064,064,064,064,064,064,064,064,064,0
    64,064,064,064,064,064
1470 data
    064,064,064,064,064,064,064,064,064,064,0
    64,064,064,064,064,064
1480 data
    064,064,064,064,064,064,064,064,125,255,2
    55,000,246,045,032,045
1490 data
    032,045,032,045,032,045,032,045,032,045,0
    32,045,032,045,032,045
1500 data
    032,045,032,045,032,045,032,045,032,045,0
    32,045,032,045,032,045
1510 data
    032,045,032,045,032,045,032,045,032,045,0
    32,045,032,045,032,045
1520 data
    032,045,032,045,032,045,032,045,032,045,0
    32,045,032,045,032,045
1530 data
    255,255,005,082,068,005,026,005,032,004,0
    05,013,015,032,012,001
1540 data
    001,020,032,004,005,032,019,014,005,012,0
    08,005,009,004,032,022
1550 data
    001,014,032,004,005,032,086,068,067,032,0
    18,001,013,032,023,018
1560 data
    009,020,005,018,032,026,009,005,014,046,0
    32,080,018,015,002,005
1570 data
    005,018,032,008,005,020,032,013,001,001,0
    18,255,005,162,005,005
1580 data
    014,019,032,009,014,032,002,001,019,009,0
    03,032,015,006,032,013
1590 data
    001,003,008,009,014,005,020,001,001,012,0
    44,032,004,001,014,032
1600 data
    026,021,012,032,010,005,032,026,009,005,0
    14,032,004,001,020,032
1610 data
    005,018,032,014,009,005,020,019,032,019,0
    14,005,012,012,005,018
1620 data
    032,009,019,032,004,001,014,255,005,242,0
    04,005,032,086,068,067
```

```
1630 data
    032,018,001,013,032,023,018,009,020,005,0
    18,046,255,255,000,086
1640 data
    045,032,045,032,045,032,045,032,045,032,0
    45,032,045,032,045,032
1650 data
    045,032,045,032,045,032,045,032,045,032,0
    45,032,045,032,045,032
1660 data
    045,032,045,032,045,032,045,032,045,032,0
    45,032,045,032,045,032
1670 data
    045,032,045,032,045,032,045,032,045,032,0
    45,032,045,032,045,032
1680 data
    045,032,045,255,255,001,068,102,102,102,1
    02,102,102,032,032,102
1690 data
    102,102,102,102,102,032,032,102,102,102,0
    32,032,102,102,102,032
1700 data
    102,102,102,032,032,102,102,102,032,032,1
    02,102,102,102,102,102
1710 data
    032,032,102,102,102,102,092,032,032,032,1
    02,102,102,102,102,102
1720 data
    032,102,102,102,102,102,032,032,032,102,1
    02,102,102,102,255,001
1730 data
    148,102,102,032,032,032,032,032,032,102,1
    02,032,032,102,102,032
1740 data
    032,102,102,032,102,102,032,102,102,032,1
    02,102,032,102,102,032
1750 data
    102,102,032,032,102,102,032,032,102,102,0
    32,032,102,102,032,032
1760 data
    102,102,032,032,102,102,032,032,102,102,0
    32,102,102,032,032,102
1770 data
    102,032,032,102,102,255,001,228,102,102,0
    32,032,032,032,032,032
1780 data
    102,102,032,032,102,102,032,032,102,102,0
    32,102,102,032,102,102
1790 data
    032,102,102,032,102,102,032,102,102,032,0
    32,102,102,032,032,102
1800 data
    102,032,032,102,102,032,032,032,102,102,0
    32,102,102,032,032,102
1810 data
    102,032,102,102,102,102,102,102,032,032,1
    02,102,102,102,255,002
1820 data
    052,102,102,032,032,032,032,032,032,102,1
    02,032,032,102,102,032
1830 data
    032,102,102,032,032,032,032,102,102,032,1
    02,102,032,032,032,032
1840 data
    102,102,032,032,102,102,032,032,102,102,0
    32,032,102,102,032,032
1850 data
    102,102,032,032,102,102,032,032,102,102,0
    32,102,102,032,102,092
1860 data
    032,032,032,102,102,255,002,132,102,102,1
    02,102,102,102,032,032
1870 data
    102,102,102,102,102,102,032,032,102,102,0
    32,032,032,032,102,102
1880 data
    032,102,102,032,032,032,032,102,102,032,0
    32,102,102,102,102,102
1890 data
    102,032,032,102,102,102,102,092,032,032,0
    32,102,102,102,102,102
1900 data
    102,032,102,102,032,032,102,102,032,032,1
    02,102,102,102,102,255
1910 data
    003,135,102,102,102,102,032,032,102,102,0
    32,032,032,102,102,032
1920 data
    032,102,102,102,102,102,032,032,102,102,1
    02,102,102,102,255,003
1930 data
    216,102,102,032,032,032,102,102,102,032,0
    32,102,102,032,032,102
1940 data
    102,032,032,032,032,032,102,102,032,032,1
    02,102,255,004,040,102
1950 data
    102,032,032,032,102,102,032,102,032,102,1
    02,032,032,102,102,102
1960 data
    092,032,032,032,102,102,032,032,102,102,2
    55,004,120,102,102,032
1970 data
    032,032,102,102,032,032,102,102,102,032,0
    32,102,102,032,032,032
1980 data
    032,032,102,102,032,032,102,102,255,004,1
    99,102,102,102,102,032
1990 data
    032,102,102,032,032,032,102,102,032,032,1
    02,102,032,032,032,032
2000 data
    032,102,102,102,102,102,102,255,255,000,1
    83,068,005,032,019,021
2010 data
    016,005,018,032,086,068,067,032,018,001,0
    13,032,023,018,009,020
2020 data
    005,018,032,004,005,013,015,255,255,007,0
    00,074,015,008,001,014
2030 data
    032,038,032,074,015,008,001,014,255,255
```

EINDE listing demo Ramwriter



PRINT OUT met o.a. Eurotopa

Hires 128

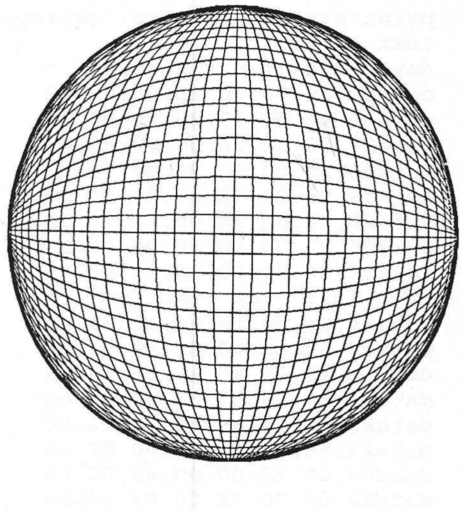
Met behulp van dit programma kunt U single color tekeningen maken. Natuurlijk alleen als U voorheen ook al een beetje kon tekenen.) U beweegt met de joystick, deze moet in poort twee zitten, het kruis dat de plaats van het volgende punt aanduidt. De kleuren zijn op de zelfde manier te wijzigen als bij basic. De maker van dit kleurige programma is Ivo Clarysse uit Waver Belgie.

```
1     rem"HIRES[SPACE]128/I.Clarysse[SPA
      CE]015/75.68.96"
10    fast:be=3583:fora=1to4:readco$:for
      b=1to64:be=be+1:readdd$:pokebe,dec
      (dd$):cc%=cc%+dec(dd$):nextb
20    ifdec(co$)<>cc%thenfl=1:ww=a:a=4
30    printhex$(cc%):cc%=0
40    nexta
50    iffl=1thenslow:print"fout[SPACE]in
      [SPACE]data,sprite[SPACE]:"ww;:sto
      p
60    clr:color0,1:color4,1:graphic0,1
70    sprite 2,1,6,0,0,0,1:sprite 3,1,6,
      0,0,0,1:sprcolor 6,3
80    movespr 2,290,91: movespr 3,290,70
90    sprite 4,1,16,0,1,0: movespr 4,40,
      70
100   kl=1
110   rem * startscherm *
120   fast:print"[SHIFT-CLR][3xCRSR-DOWN
      ][CTRL-8]"spc(16)"HIRES[SPACE]128"
      :printspc(15)"-----------":print"[
      4xCRSR-DOWN][CTRL-N]"
130   print"[4xSPACE]Met[SPACE]dit[SPACE
      ]programma[SPACE]kan[SPACE]je[SPAC
      E]single":print"[4xSPACE]color[SPA
      CE]tekeningen[SPACE]maken[SPACE]."
      :print"[4xSPACE]U[SPACE]beweegt[SP
      ACE]met[SPACE]uw[SPACE]joystick[SP
      ACE](poort[SPACE]2)"
140   print"[4xSPACE]het[SPACE]kruis[SPA
      CE]dat[SPACE]de[SPACE]plaats[SPACE
      ]van[SPACE]het":print"[4xSPACE]vol
      gende[SPACE]punt[SPACE]aanduidt[SP
      ACE].":print"[4xSPACE]De[SPACE]kle
      uren[SPACE]kan[SPACE]men[SPACE]op[
      SPACE]dezelfde"
150   print"[4xSPACE]wijze[SPACE]wijzige
      n[SPACE]als[SPACE]in[SPACE]BASIC[S
      PACE].":print"[4xSPACE]Druk[SPACE]
      op[SPACE]'H'[SPACE]voor[SPACE]help
      [SPACE]informatie.":print"[4xSPACE
      ]Dit[SPACE]programma[SPACE]is[SPAC
      E]gemaakt[SPACE]door"
160   print"[3xSPACE]GDPIC[SPACE]softwar
      e":print"[3xSPACE]Oude[SPACE]bergs
      traat[SPACE]8":print"[3xSPACE]2860
      [SPACE]Oude[SPACE]Bergstraat[SPACE
      ]8":print"[3xSPACE]BELGIE":print"[
      3xSPACE]015/75.68.96":slow:getkeyy
      $
170   graphic 1,1:sprite2,0:sprite3,0:sp
      rite4,0:color0,1:color4,1:color1,6
      :x=0:y=0:sprite1,1,16,0,0,0,0
180   rem * hoofdroutine *
190   movespr1,(x+14),y+40
200   gett$:ift$<>""thengosub240
210   ifjoy(2)<>0thenj=joy(2):gosub380
220   goto190
230   rem  ** letter invoer **
240   ift$="[SHIFT-CLR]"thenscnclr3:retu
      rn
250   ift$="0"thenkl=0:return
260   ift$="1"thenkl=1:return
290   ift$="p"thenpaintkl,x,y:return
300   ift$="h"thengosub480:return
310   ift$="+"thenxx%=x:yy%=y:return
320   ift$="-"thendrawkl,xx%,yy%tox,y:re
      turn
330   ift$="r"thenboxkl,xx%,yy%,x,y:retu
      rn
340   ift$="c"thencirclekl,xx%,yy%,abs(x
      x%-x),abs(xx%-x):return
350   ift$="d"thengosub680:return
360   printt$:colorkl,rclr(5):return
370   rem ** joystick invoer **
380   f=0:ifj>127thenf=1:j=j-128
390   ifj=1orj=8orj=2theny=y-1
400   ifj=4orj=5orj=6theny=y+1
410   ifj=6orj=7orj=8thenx=x-1
420   ifj=2orj=3orj=4thenx=x+1
430   ifx<0thenx=0:elseifx>319thenx=319
440   ify<0theny=0:elseify>199theny=199
450   iff=0thenreturn
460   drawkl,x,y:return
470   rem *** help ***
480   graphic0:print"[SHIFT-CLR][CTRL-N]
      ":sprite1,0
490   print"[2xSPACE]HIRES[SPACE]128[SPA
      CE]....[SPACE]help"
500   print
510   print"[2xSPACE]clr..wis[SPACE]het[
      SPACE]scherm"
520   print"[2xSPACE]0....werk[SPACE]met
      [SPACE]de[SPACE]achtergrondkleur"
530   print"[2xSPACE]1....werk[SPACE]met
      [SPACE]de[SPACE]voorgrondkleur"
560   print"[6xSPACE][COM-A]kies[SPACE]b
      eginpunt[SPACE]lijn"
570   print"[2xSPACE]+...[COM-Q]kies[SPA
      CE]middelpunt[SPACE]cirkel"
580   print"[6xSPACE][COM-Z]kies[SPACE]l
      inkerbovenhoek[SPACE]rechthoek"
590   print"[2xSPACE]-....kies[SPACE]ein
      dpunt[SPACE]lijn"
600   print"[2xSPACE]C....teken[SPACE]ee
      n[SPACE]cirkel"
610   print"[2xSPACE]R....teken[SPACE]ee
      n[SPACE]rechthoek"
620   print"[2xSPACE]P....vul[SPACE]een[
      SPACE]vlak[SPACE]op"
630   print"[2xSPACE]D....disketteoperat
      ies"
640   print
650   print"[SPACE][CTRL-9][SPACE]druk[S
      PACE]op[SPACE]een[SPACE]toets[SPAC
      E]om[SPACE]terug[SPACE]te[SPACE]ke
      ren[SPACE][CTRL-0]"
660   getkeyy$:sprite1,1:graphic1:return
670   rem * disketteoperaties *
680   graphic0:sprite1,0:print"[SHIFT-CL
      R][CTRL-N][SPACE]HIRES[SPACE]128[S
      PACE]....[SPACE]disketteoperaties"
690   print:print"[SPACE]1[SPACE]==>[SPA
      CE]directory":print"[SPACE]2[SPACE
      ]==>[SPACE]lezen":print"[SPACE]3[S
      PACE]==>[SPACE]schrijven":print"[S
      PACE]4[SPACE]==>[SPACE]formatteren
      "
700   print"[SPACE]6[SPACE]==>[SPACE]ter
```

```
    ug[SPACE]naar[SPACE]het[SPACE]werk
    scherm"
710 print"[SPACE][CTRL-9][SPACE]geef[S
    PACE]uw[SPACE]keuze[SPACE]:[SPACE]
    .[2xSPACE][CTRL-0]"
720 window0,24,39,24,1
730 getkeyy$:w=val(y$):ifw<1orw>6then7
    30
740 poke1322,176+w
750 ifw=6thensprite1,1:graphic1:window
    0,0,39,24:return
760 onwgosub790,820,910,1020,1080
770 poke1322,174:goto720
780 rem * directory *
790 window0,9,39,23,1:catalog"dd*":ifd
    s>19thengosub1120
800 return
810 rem * lezen *
820 input"bestandsnaam[SPACE]";l$:ifl$
    =""thenreturn
830 l$="dd"+left$(l$,14)
840 bload(l$),b0,p7168
890 ifds>19thengosub1120:return:elsere
    turn
900 rem * schrijven *
910 input"bestandsnaam[SPACE]";l$:ifl$
    =""thenreturn
920 l$="dd"+left$(l$,14)
950 bsave(l$),b0,p7168top16384
990 ifds>19thengosub1120
1000 return
1010 rem * wissen *
1020 input"bestandsnaam[SPACE]";l$:ifl$
     =""thenreturn
1030 l$="dd"+left$(l$,14)
1040 scratch(l$)
1050 ifds>19thengosub1120
1060 return
1070 rem * formatteren *
1080 print"formatteren[SPACE]?";:getkey
     y$:ify$<>"j"thenreturn
1090 header(chr$(34)+"[CTRL-0]gdpic[SPA
     CE]software"),igi
1100 ifds>19thengosub1120:return:elsere
     turn
1110 rem * disk foutmeldingen *
1120 print"st:"st",ds:"ds$;:getkeyy$:re
     turn
1130 data 0516 :rem * sprite 1 *
1140 data00,00,00,00,20,00,00,20
1150 data00,00,20,00,00,20,00,00
1160 data20,00,00,20,00,00,20,00
1170 data00,20,00,00,20,00,ff,df
1180 dataf8,00,20,00,00,20,00,00
1190 data20,00,00,20,00,00,20,00
1200 data00,20,00,00,20,00,00,20
1210 data00,00,20,00,00,00,00,00
1220 data 189f :rem * sprite 2 *
1230 datafe,aa,a0,fe,aa,a0,3e,aa
1240 dataa8,3f,aa,a0,3f,aa,a8,0f
1250 dataaa,a8,0f,aa,a8,0f,ea,a8
1260 data03,ea,aa,03,ea,a8,00,ea
1270 dataaa,00,ea,a8,00,2a,aa,00
1280 data2a,a8,00,0a,aa,00,02,aa
1290 data00,00,a2,00,00,a2,00,00
1300 data22,00,00,22,00,00,22,00
1310 data 2214 :rem * sprite 3 *
1320 data0a,a0,00,2a,a8,00,20,aa
1330 data00,8c,2a,00,f0,2a,00,c0
1340 data0a,80,c0,2a,80,c0,aa,80
1350 data02,aa,80,2a,aa,a0,3a,aa
1360 dataa0,3a,aa,a0,3a,aa,a0,fa
1370 dataaa,a0,fa,aa,a0,fa,aa,a0
1380 datafe,aa,a0,fe,aa,a0,fe,aa
1390 dataa0,fe,aa,a0,fe,aa,a0,00
1400 data 1217 :rem * sprite 4 *
1410 data29,28,00,2b,b8,00,12,a8
1420 data00,00,00,00,6c,c9,00,8a
1430 dataaa,00,aa,ca,00,aa,8a,00
1440 data4c,89,00,00,00,00,52,27
1450 data00,55,56,00,22,25,00,00
1460 data00,00,92,73,80,9a,46,70
1470 data96,44,60,92,64,00,92,44
1480 data60,92,46,70,92,43,80,00
```

** EINDE LISTING hires 128 **

regel 1	223	regel 320	150	regel 640	153	regel 1030	178	regel 1330	33
regel 10	209	regel 330	63	regel 650	238	regel 1040	179	regel 1340	59
regel 20	202	regel 340	62	regel 660	245	regel 1050	53	regel 1350	95
regel 30	46	regel 350	215	regel 670	236	regel 1060	142	regel 1360	125
regel 40	195	regel 360	16	regel 680	233	regel 1070	42	regel 1370	158
regel 50	193	regel 370	122	regel 690	197	regel 1080	88	regel 1380	192
regel 60	161	regel 380	24	regel 700	80	regel 1090	24	regel 1390	132
regel 70	222	regel 390	97	regel 710	48	regel 1100	154	regel 1400	118
regel 80	94	regel 400	100	regel 720	97	regel 1110	223	regel 1410	22
regel 90	211	regel 410	105	regel 730	28	regel 1120	181	regel 1420	5
regel 100	122	regel 420	92	regel 740	42	regel 1130	116	regel 1430	90
regel 110	51	regel 430	53	regel 750	29	regel 1140	187	regel 1440	239
regel 120	82	regel 440	69	regel 760	133	regel 1150	187	regel 1450	215
regel 130	21	regel 450	232	regel 770	131	regel 1160	189	regel 1460	255
regel 140	53	regel 460	77	regel 780	152	regel 1170	17	regel 1470	252
regel 150	73	regel 470	180	regel 790	220	regel 1180	217	regel 1480	240
regel 160	68	regel 480	146	regel 800	142	regel 1190	189		
regel 170	163	regel 490	84	regel 810	97	regel 1200	189		
regel 180	121	regel 500	153	regel 820	69	regel 1210	185		
regel 190	172	regel 510	176	regel 830	178	regel 1220	145		
regel 200	200	regel 520	33	regel 840	192	regel 1230	173		
regel 210	200	regel 530	177	regel 890	154	regel 1240	134		
regel 220	35	regel 560	146	regel 900	143	regel 1250	152		
regel 230	218	regel 570	25	regel 910	69	regel 1260	106		
regel 240	22	regel 580	185	regel 920	178	regel 1270	77		
regel 250	17	regel 590	130	regel 950	185	regel 1280	58		
regel 260	19	regel 600	97	regel 990	53	regel 1290	221		
regel 290	55	regel 610	83	regel 1000	142	regel 1300	195		
regel 300	217	regel 620	1	regel 1010	188	regel 1310	115		
regel 310	142	regel 630	98	regel 1020	69	regel 1320	41		



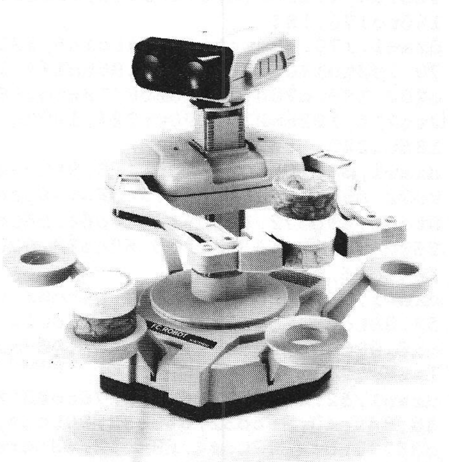
Topografiespel voor 128

Deze keer een inzendster we krijgen natuurlijk wel eens meer inzendingen van dames maar of er wordt alleen een voorletter vermeld en dan kunnen we natuurlijk niet zien dat het een vrouwelijke inzender is. Hier was het echter wel duidelijk de schrijf**(ster)** van dit programma is Marian van Hal Kouwenberg. uit Drunen. Topografie van europa op je computer. Leerzaam en leuk.

```
10    rem ******************************
      ***
20    rem **
        **
30    rem ******* e u r o t o p a ******
      ***
40    rem **
        **
50    rem ******* commodore   128 ******
      ***
60    rem **
        **
70    rem ******* marian van  hal ******
      ***
80    rem **
        **
90    rem ******* drunen   1 9 8 7 ******
      ***
100   rem**
      **
110   rem*******************************
      **
120   color 0,1:color 4,1:scnclr:for xx=
      1 to 23: print:next xx
130   print"[CTRL-8][4xCOM-+][SPACE][COM
       +][2xSPACE][COM-+][SPACE][4xCOM-+
       ][2xSPACE][2xCOM-+][2xSPACE][5xCOM
       +][2xSPACE][2xCOM-+][2xSPACE][4xC
      OM +][2xSPACE][2xCOM-+]":sleep 1:p
      rint"[COM-+][4xSPACE][COM-+][2xSPA
      CE][COM-+][SPACE][COM-+][2xSPACE][
      COM-+][SPACE][COM-+][2xSPACE][COM-
      +][3xSPACE][COM-+][3xSPACE][COM-+]
      [2xSPACE][COM-+][SPACE][COM-+][2xS
      PACE][COM-+][SPACE][COM-+][2xSPACE
      ][COM-+]":sleep 1:print"[COM-+][4x
      SPACE][COM-+][2xSPACE][COM-+][SPAC
      E][COM-+][2xSPACE][COM-+][SPACE][C
      OM +][2xSPACE][COM-+][3xSPACE][COM
       +][3xSPACE][COM-+][2xSPACE][COM-+
      ][SPACE][COM-+][2xSPACE][COM-+][SP
      ACE][COM-+][2xSPACE][COM-+]"
140   sleep 1:print"[3xCOM-+][2xSPACE][C
      OM +][2xSPACE][COM-+][SPACE][4xCOM
       +][SPACE][COM-+][2xSPACE][COM-+][
      3xSPACE][COM-+][3xSPACE][COM-+][2x
      SPACE][COM-+][SPACE][4xCOM-+][SPAC
      E][COM-+][2xSPACE][COM-+]":sleep 1
      :print"[COM-+][4xSPACE][COM-+][2xS
      PACE][COM-+][SPACE][2xCOM-+][3xSPA
      CE][COM-+][2xSPACE][COM-+][3xSPACE
      ][COM-+][3xSPACE][COM-+][2xSPACE][
      COM-+][SPACE][COM-+][4xSPACE][4xCO
      M +]":sleep 1:print"[COM-+][4xSPAC
      E][COM-+][2xSPACE][COM-+][SPACE][C
      OM +][SPACE][COM-+][2xSPACE][COM-+
      ][2xSPACE][COM-+][3xSPACE][COM-+][
      3xSPACE][COM-+][2xSPACE][COM-+][SP
      ACE][COM-+][4xSPACE][COM-+][2xSPAC
      E][COM-+]"
150   sleep 1:print "[4xCOM-+][2xSPACE][
      3xCOM-+][SPACE][COM-+][2xSPACE][CO
      M +][2xSPACE][2xCOM-+][4xSPACE][CO
      M +][4xSPACE][2xCOM-+][2xSPACE][CO
      M +][4xSPACE][COM-+][2xSPACE][COM-
      +]"
160   for xx=1 to 6: print"[CRSR-DOWN]";
      :sleep 1:next xx:print"[3xCRSR-RIG
      HT]een[SPACE]topografiespel":sleep
       1
170   for xx=1 to 4: print"[CRSR-DOWN][4
      xCRSR-RIGHT]";:sleep 1: next xx:pr
      int"van":sleep 1
180   for xx= 1 to 4:print"[CRSR-DOWN][5
      xCRSR-RIGHT]";:sleep 1:next xx:pri
      nt"marian[SPACE]van[SPACE]hal
190   sleep 6
200   for xx=1 to 23: print : next xx
210   print"je[SPACE]komt[SPACE]met[SPAC
      E]een[SPACE]ruimteschip[SPACE]van[
      SPACE]mars[SPACE]en[SPACE]moet[SPA
      CE]europa[SPACE]verkennen.[SPACE]h
      elaas[SPACE]weet[SPACE]je[4xSPACE]
      niet[SPACE]waar[SPACE]alles[SPACE]
      ligt."
220   print"[CRSR-DOWN]je[SPACE]bestuurt
      [SPACE]je[SPACE]ruimteschip[SPACE]
      met[SPACE]de[SPACE]joy-[2xSPACE]st
      ick;[SPACE]als[SPACE]je[SPACE]denk
      t[SPACE]dat[SPACE]je[SPACE]boven[S
      PACE]de[5xSPACE]juiste[SPACE]plaat
      s[SPACE]gearriveerd[SPACE]bent[SPA
      CE]kun[SPACE]je[3xSPACE]landen[SPA
      CE]d.m.v.[SPACE]de[SPACE]vuurknop.
      "
230   print"[CRSR-DOWN]na[SPACE]de[SPACE
      ]landing[SPACE]krijg[SPACE]je[SPAC
      E]een[SPACE]nieuwe[7xSPACE]opdrach
      t;[SPACE]stijg[SPACE]op[SPACE]en[S
      PACE]voer[SPACE]dezelfde[SPACE]ma-
      [SPACE]noeuvres[SPACE]uit.[SPACE]a
      ls[SPACE]je[SPACE]10[SPACE]keer[SP
      ACE]goed[SPACE]gelandbent[SPACE]kr
      ijg[SPACE]je[SPACE]een[SPACE]bonus
      ";
240   print"[SPACE]en[SPACE]kun[SPACE]je
      [SPACE]weer[2xSPACE]kiezen[SPACE]u
```

```
     it[SPACE]de[SPACE]diverse[SPACE]mo
     gelijkheden."
250  print"[CRSR-DOWN]je[SPACE]kunt[SPA
     CE]stoppen[SPACE]door[SPACE]op[SPA
     CE]run-stop/restorete[SPACE]drukke
     n."
260  print"joystick[SPACE]in[SPACE]poor
     t[SPACE]2:[SPACE]start[SPACE]d.m.v
     .[SPACE]vuur-[SPACE]knop"
270  if joy(2)<128then 270
280  color 0,4:color 4,4:color 1,4:grap
     hic1,1
289  rem
290  rem **** sprites ontwerpen ****
291  rem
300  circle1,12,10,6,2,,,340:circle1,11
     ,8,3,3,230,50,20:paint 1,12,10:dra
     w0,9,11:draw0,12,10:draw0,15,9
310  shape a$,1,1,24,21
320  sprsave a$,1:graphic1,1
330  circle 1,3,3,1,1:paint 1,3,3
340  shape b$,1,1,24,21
350  sprsave b$,2:sprsave2,3:sprsave 3,
     4:sprsave4,5:sprsave5,6:sprsave6,7
     :sprsave7,8
359  rem
360  rem **** europa tekenen ****
361  rem
370  graphic 1,1:color 0,4:color 4,7:co
     lor 1,7
380  draw 1,188,30to188,40to240,40to 24
     0,88 to 232,92 to 228,96 to 228,10
     4 to 224,104 to 224,108 to 240,112
      to 240,186
390  draw1,111,190to102,190to111,198to1
     11,190:draw1,88,184to88,174to94,17
     4to90,182to88,184:draw1,92,172to94
     ,164to90,166to90,172to92,172:rem**
     eil.midd.zee****
400  draw 1,36,0to 48,0to 48,4 to 56,
     4 to56,8 to 60,8 to 60,12 to 56,12
      to 56,16 to 40,16 to 40,12 to 36,
     12 to 36,0
410  draw 1,136,2to144,2to144,0to156,0t
     o156,4to160,4to160,2to168,2to168,4
     to174,4to184,4to184,8to192,8to192,
     12to196,12to196,16to192,16to192,20
     to184,20to184,16
420  draw1,184,16to176,16to184,24to184,
     32to188,32to188,28to196,28to196,24
     to198,24to198,16to204,16to204,12to
     200,12to200,8to198,8to198,0to204,0
     to204,8to210,8
430  draw1,210,8to210,0to240,0
440  draw1,136,2to136,8to132,8to132,12t
     o128,12to128,16to120,24to112,32to1
     12,36to108,36to108,40to104,40to96,
     48to96,60to92,60to92,64to96,64to96
     ,68to108,68to108,64
450  draw1,108,64to112,64to112,80to120,
     80to120,88to124,88to124,84to128,84
     to128,72to136,64to132,64to132,48to
     128,48to144,32to144,26to152,26to15
     2,32to144,40
460  draw 1,144,40to143,40to143,56to144
     ,56to144,59to160,59to160,56to168,5
     6to168,62to160,62to160,64to151,64t
     o151,72to144,72to144,76to148,76to1
     48,80to143,80
470  draw 1,143,80to143,88to144,88to136
     ,96to136,93to128,93to128,96to116,9
     6to116,92to112,92to112,93to104,93t
     o104,84to107,84to107,88to110,88to1
     10,84to108,80
480  draw1,108,80to108,72to104,72to104,
     76to100,76to100,80to98,80to98,88to
     100,88to100,96to88,96to80,104to80,
     105to72,105to72,108to64,108to64,11
     2to60,112to60,108
490  draw1,60,108to56,108to56,114to48,1
     14to48,112to44,112to44,116to48,116
     to48,120
500  draw1,48,120to52,120to52,136to50,1
     36to50,144to40,144to40,140to32,140
     to32,138to24,138to24,136to16,136to
     16,144to15,144to15,152to16,152to8,
     160to8,170to16,170
510  draw1,16,170to16,178to24,178to24,1
     76to32,176to32,178to40,178to40,176
     to44,176to44,174to48,174to48,168to
     52,168to52,164to56,164to56,162to64
     ,162to64,160
520  draw1,64,160to66,160to66,152to80,1
     52to80,155to88,155to88,152to100,15
     2to100,164to104,164to104,168to108,
     168to108,172to116,172to116,180to12
     0,180to120,184
530  draw1,120,184to114,190to117,193to1
     23,187to123,180to126,180to126,182t
     o128,182to128,174to124,174to122,17
     2to122,168to114,160to114,156to110,
     156to110,148
540  draw1,110,148to114,148to114,155to1
     22,155to122,160to128,160to128,164t
     o132,164to132,168to136,168to136,17
     6to139,176to139,184to140,184to140,
     188to144,188
550  draw1,144,188to144,192to152,192to1
     52,196to160,196to152,188to152,184t
     o156,184to156,176to164,176to164,17
     8to172,178to172,174to170,174to170,
     166to168,166
560  draw1,168,166to168,162to172,162to1
     72,154to176,154to176,146to184,146t
     o184,150to188,150to188,154to192,15
     4to192,150to196,150to196,146to200,
     146to200,150
570  draw1,200,150to216,156to224,156to2
     24,168to220,168to220,172to200,172t
     o200,168to192,168to192,164to180,16
     4to180,176to176,176to176,180to170,
     180to170,184
580  draw1,170,184to166,184to166,192to1
     70,192to170,196to174,196to174,199t
     o200,199to200,196to208,196to208,19
     2to216,192to216,190to224,190to224,
     186to240,186
590  draw1,64,100to68,100to68,96to72,96
     to72,88to68,88to68,76to66,76to66,6
     8to64,68to72,60to64,60to64,56to68,
     56to68,52to64,52to56,60to52,60to52
     ,64to56,64to56,72
600  draw1,56,72to60,72to60,84to52,84to
     52,88to48,88to48,94to44,94to44,98t
     o56,98to56,96to64,96to64,100:draw
     1,56,72 to 66,68
610  draw1,52,68to52,76to48,76to48,84to
     40,84to40,86to32,86to32,80to36,80t
     o36,72to44,72to44,68to52,68:draw1,
     44,72to48,76
620  draw1,240,40to240,199:color 1,7:pa
     int1,6,6:paint1,190,22:paint 1,112
```

```
     ,150:paint 1,206,2:paint1,238,90:p
     aint 1,238,190
627  rem
628  rem***** landen intekenen ****
629  rem
630  drawl,112,65to116,55to116,43to122,
     40to122,32to134,20to142,12to146,12
     to146,0:rem noorwegen*********
640  drawl,146,0to150,12to150,26:rem***
     *zweden****
650  drawl,164,56to172,46to172,36to164,
     18to164,8to160,4:rem****finland
660  drawl,166,62to162,68to162,74to152,
     72:drawl,162,74to170,82to164,94tol
     58,98to142,92:rem **** baltische s
     taten ****
670  drawl,158,98to158,104to162,112to15
     4,120to140,120to140,116to128,116to
     120,108to116,96:rem **** polen ***
     *
680  drawl,120,108to105,112 to105,108to
     100,106to108,94:drawl,108,111to115
     ,124to112,128to112,130to100,130to8
     8,126to92,120to88,110to94,96:rem**
     **o+w-duitsland****
690  drawl,88,110to84,110to86,106to82,1
     02:rem****nederland****
700  drawl,76,106to78,112to88,114:drawl
     ,84,110to84,113:rem **** belg/lux
     ****
710  drawl,88,126to84,126to84,140to82,1
     44to86,152:drawl,66,152to50,144:re
     m **** frankrijk/spanje****
720  draw 1,16,144to24,146to16,170:rem*
     *** portugal ****
730  drawl,84,140to100,136to106,136toll
     2,142to112,148:drawl,112,142to126,
     142to130,136to130,126to122,122toll
     5,124:rem**** italie/zwitserl. ***
     *
740  drawl,126,142to132,146to142,144tol
     48,150to152,150to152,170to144,170t
     o140,164to136,168:drawl,145,170tol
     40,180:rem **** joegoslavie/albani
     e
750  drawl,152,170to164,169to164,176:dr
     awl,152,160to166,156to172,160:draw
```



```
     1,164,172to170,172     :rem **** gr
     iekenl/bulg ****
760  drawl,130,132to154,132to154,120:dr
     awl,154,132to166,132to170,128to170
     ,146to180,146:drawl,142,144to156,1
     32:rem **** rest ****
770  drawl,100,90to104,90:rem**** denem
     arken ****
780  drawl,100,130to100,136
789  rem
790  rem***** keuzeroutine ***********
791  rem
800  a1$="je[SPACE]kunt":a2$="kiezen":a
     3$="uit:...":a4$="[COM-+]landen":a
     5$="[COM-+]plaatsen":a6$="[COM-+]s
     treken+":a7$="zeeen":a8$="kies[SPA
     CE]met":a9$="joystick":b1$="en":b2
     $="vuurknop"
810  char 1,31,2,a1$:char1,31,3,a2$:cha
     r 1,31,4,a3$:char1,31,6,a4$:char 1
     ,31,8,a5$:char 1,31,10,a6$:char 1,
     31,11,a7$
820  char 1,31,14,a8$:char 1,31,15,a9$:
     char 1,31,16,b1$:char 1,31,17,b2$
830  if z1=1 then z1=0:b3$="[9xSPACE]":
     for i=1to17:char1,31,i,b3$:next i:
     sprite2,0:restore v:goto 910
840  sprite 2,1,1,0,1,1:x=270:y=150:mov
     espr2,x,y
850  if joy(2)=1theny=y-1
860  ifjoy(2)=5 theny=y+1
870  if joy(2)=>128 and y>95andy<99then
      z=32:z1=1:v=1000:goto 830
880  if joy(2)=>128 and y>111andy<115th
     en z=30:z1=1:v=1050:goto 830
890  if joy(2)=>128 and y>127andy<131 t
     hen z=26:z1=1:v=1100:goto 830
900  movespr2,x,y:goto 850
907  rem
908  rem **** kaart intekenen *****
909  rem
910  color 1,1:if z=26 or z=32 then q=1
     :else q=0
920  sprite 1,1,8,0,0,0,0:movespr1,40,5
     0
930  char 1,25,2,"[15xSPACE]"
940  sprite 2,1,1,0,q,q,0:sprite 3,1,1,
     0,q,q,0:sprite4,1,1,0,q,q,0:sprite
      5,1,1,0,q,q,0:sprite6,1,1,0,q,q,0
     :sprite 7,1,1,0,q,q,0:sprite 8,1,1
     ,0,q,q,0
950  fast:sound 2,20000,160,2,2000,100:
     k=int(rnd(1)*7)+1:for h=2 to 8:j=i
     nt(rnd(1)*z)+1:for i=1toz:read f,g
     ,l$:ifh=k+1and j=i then l1$=l$:p1=
     f:p2=g
960  if j=i then movesprh,f,g
970  next i:restorev :next h:slow
980  char 1,31,20,"[6xSPACE]"
990  le=len(l1$):if le>9 then char 1,31
     -(le-9),2,l1$:else char 1,31,2,l1$
1000 rem**plaatsbepalingen*landen******
     *
1010 data 66,54,ijsland,84,129,engeland
     ,62,124,ierland,33,204,portugal,57
     ,204,spanje,88,183,frankrijk,106,1
     58,luxemburg,100,155,belgie,107,14
     7,nederland
1020 data 116,159,w-duitsland,131,147,o
     -duitsland,122,126,denemarken,127,
     92,noorwegen,142,101,zweden,176,91
```

```
     ,finland,223,123,rusland,175,113,e
     stland
1030 data 175,122,letland,175,134,litou
     wen,157,152,polen,140,163,t-slowak
     ije,140,181,oostenrijk,113,179,zwi
     tserland,123,195,italie,153,201,jo
     egoslavie
1040 data 160,217,albanie,168,225,griek
     enland,207,225,turkije,180,210,bul
     garije,180,189,roemenie,157,184,ho
     ngarije,83,110,schotland
1050 rem **** pl. bep. steden *********
1060 data61,59,reykjavik,71,119,belfast
     ,68,127,dublin,89,141,londen,33,20
     9,lissabon,57,207,madrid,91,166,pa
     rijs,103,155,brussel,110,148,amste
     rdam
1070 data 109,160,luxemburg,132,109,osl
     o,153,115,stockholm,174,105,helsin
     ki,131,132,kopenhagen,134,148,berl
     ijn,115,159,bonn,112,180,bern,133,
     217,rome
1080 data 173,237,athene,192,224,istanb
     oel,164,200,belgrado,161,219,tiran
     a,179,212,sofia,187,199,boekarest,
     160,184,boedapest,147,176,wenen,13
     9,162,praag
1090 data 168,155,warschau,214,130,mosk
     ou,80,113,glasgow
1100 rem **** pl. bep. streken/zeeen***
1110 data 212,69,barentszee,169,50,noor
     dkaap,166,56,lapland,38,100,atl.oc
     eaan,102,131,noordzee,90,149,het k
     anaal,79,158,normandie,69,161,bret
     agne
1120 data 60,176,golf v.biskaje,76,195,
     pyreneeen,40,222,gibraltar,97,222,
     middellandse z.,146,212,adriatisch
     e zee,152,236,ionische zee,111,214
     ,corsica
1130 data 128,238,sicilie,182,230,egeis
     che zee,218,203,zwarte zee,193,226
     ,bosporus,254,147,kaspische zee,12
     6,116,skagerrak,132,122,kattegat,1
     55,128,oostzee
1140 data 157,95,botnische golf,175,112
     ,finse golf,109,223,sardinie
1147 rem
1148 rem ***** joystick besturing ***
1149 rem
1150 sleep 1:sound2,20000,100,0,2000,10
     0
1160 if joy(2)=0then b=0:else b=1
1170 if joy(2)=1 then a=0
1180 if joy(2)=2 then a=45
1190 if joy(2)=3 then a=90
1200 if joy(2)=4 then a=135
1210 if joy(2)=5 then a=180
1220 if joy(2)=6 then a=225
1230 if joy(2)=7 then a=270
1240 if joy(2)=8 then a=315
1250 if joy(2)=>128thenge=1:c=rsppos(1,
     0):d=rsppos(1,1):goto 1280
1260 if joy(2)<>0then sound 3, 6000,1
1270 movespr1,a#b:goto 1160
1280 ifc>p1+10orc<p1-10then goto 1310:e
     lse1290
1290 if d>p2+10ord<p2-10then goto 1310
     :else p=p+1:pp=pp+1:p$=str$(pp):ch
     ar 1,33,23,p$
1300 if p=10then p=0:pp=pp+5:p$=str$(p
     p):char1,33,23,p$:goto 800:else go
     to 930
1310 char 1,31,20,"fout":sleep 2:goto 9
     30
```

** EINDE LISTING eurotopa **

regel 10	249	regel 291	143	regel 570	218	regel 820	150	regel 1090	149
regel 20	55	regel 300	64	regel 580	10	regel 830	41	regel 1100	70
regel 30	158	regel 310	36	regel 590	133	regel 840	56	regel 1110	224
regel 40	55	regel 320	124	regel 600	96	regel 850	167	regel 1120	165
regel 50	111	regel 330	147	regel 610	11	regel 860	170	regel 1130	8
regel 60	55	regel 340	37	regel 620	141	regel 870	80	regel 1140	3
regel 70	161	regel 350	48	regel 627	143	regel 880	157	regel 1147	143
regel 80	55	regel 359	143	regel 628	92	regel 890	163	regel 1148	2
regel 90	212	regel 360	181	regel 629	143	regel 900	159	regel 1149	143
regel 100	55	regel 361	143	regel 630	149	regel 907	143	regel 1150	98
regel 110	249	regel 370	138	regel 640	182	regel 908	29	regel 1160	190
regel 120	222	regel 380	146	regel 650	154	regel 909	143	regel 1170	138
regel 130	239	regel 390	181	regel 660	109	regel 910	236	regel 1180	196
regel 140	135	regel 400	67	regel 670	220	regel 920	247	regel 1190	197
regel 150	87	regel 410	105	regel 680	3	regel 930	114	regel 1200	246
regel 160	15	regel 420	14	regel 690	134	regel 940	230	regel 1210	247
regel 170	19	regel 430	98	regel 700	237	regel 950	78	regel 1220	248
regel 180	39	regel 440	64	regel 710	235	regel 960	168	regel 1230	249
regel 190	63	regel 450	84	regel 720	149	regel 970	73	regel 1240	250
regel 200	92	regel 460	117	regel 730	223	regel 980	159	regel 1250	65
regel 210	212	regel 470	135	regel 740	70	regel 990	247	regel 1260	116
regel 220	65	regel 480	152	regel 750	53	regel 1000	127	regel 1270	146
regel 230	76	regel 490	243	regel 760	171	regel 1010	121	regel 1280	14
regel 240	87	regel 500	147	regel 770	41	regel 1020	109	regel 1290	157
regel 250	81	regel 510	143	regel 780	142	regel 1030	122	regel 1300	132
regel 260	154	regel 520	111	regel 789	143	regel 1040	122	regel 1310	177
regel 270	107	regel 530	190	regel 790	217	regel 1050	67		
regel 280	132	regel 540	227	regel 791	143	regel 1060	15		
regel 289	143	regel 550	26	regel 800	21	regel 1070	34		
regel 290	203	regel 560	245	regel 810	36	regel 1080	202		

Commodore heeft ontdekt, dat de Amiga heel goed te gebruiken is als produktiesysteem voor televisiestations. Vooral bij kleinere organisaties kan men er effectief ondertiteling, kaarten en graphics mee maken en met live beelden mengen.

Amiga te huur ...
...als televisiestudio

De Sport marketing afdeling van Commodore Europa in Frankfurt probeert op dit moment aan de diverse omroeporganisaties een complete produktieploeg, compleet met Amiga's, snijapparatuur, en andere videospullen te slijten voor 12.000 DM per dag. Men heeft er zelfs een speciale Amiga tele Mobiel voor gemaakt, een hele truck met spullen.

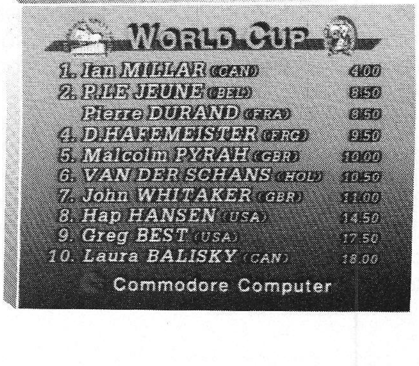
Amiga Tele-Vision

Met dit motto gaat men nu vooral de sportclubs interesseren in het maken van video-verslagen van hun wedstrijden. Die kan men dan weer doorverkopen aan de omroepen, of ook voor eigen publiek nog eens gebruiken, er banden van verkopen en dergelijke. Om daar allemaal een exclusieve touch aan te geven, worden de Amiga's dan gebruikt om extra dingen te doen.



Golf

Een goed voorbeeld is een golfwedstrijd. het is vrij moeilijk om de informatie over wat er op de verschillende holes gebeurt, goed aan de kijker over te brengen. Met behulp van getekende kaartjes en grafische informatie over de stand kan dat veel duidelijker worden. De Amiga wordt in zo'n geval gebruikt om al die extra te maken en in beeld te krijgen via de Genlock techniek.



Multi-media

Commodore probeert haar multimedia mobiel te verhuren voor dergelijke evenementen. Men heeft in het verleden door de eigen sponsoring van bijvoorbeeld voetbalteams veel ervaring opgedaan en dat wordt nu goed benut. Er is in Frankfurt zelfs een speciale sport-marketing divisie voor opgericht. Men verwacht erg veel van deze sector, getuige de promotiefolder, die men bij onze oosterburen uitgebracht heeft.

Toekomst

Toch is het niet de bedoeling, dat de Truck uitsluitend bij sportevenementen wordt ingezet. Op langere termijn verwacht het concern, dat ook andere organisatoren van de mogelijkheden gebruik zullen maken. Tijdens eenmalige bijzondere gebeurtenissen, die live worden uitgezonden kan de bus uitstekende diensten bewijzen. Te denken valt aan popfestivals, shows en demonstraties. Maar er kunnen tegen een relatief lage kostprijs ook videofilms vervaardigd worden van bedrijven of produkten, die door eenvoudige montagemogelijkheden voorzien kunnen worden van grafieken, data of andere informatie.

Wie nadere informatiewil, kan zich wenden tot H. Hocke, int. Sports Promotion; Commodore Electronics Ltd. (Duitsland). Tel.: 069-66 38-175

De taal C wordt vaak gezien als de programmeertaal van de toekomst, maar het is eigenlijk al weer een programmeertaal uit het verleden. Wat het ook zal zijn, voor de Commodore Amiga is het DE taal om mee te werken. Wij zullen in deze cursus proberen ook de BASIC programmeur een C-kans op de Amiga te geven. Veel geduld en een goede C-compiler zult u zondermeer moeten hebben



De oorsprong van C

Amiga C cursus, deel 1

Zoals de kop al deed vermoeden, bestaat de taal C al weer geruime tijd. De wortels van C liggen ergens in de late jaren '60. In 1967 werd een taal ontwikkeld die BCPL was genaamd. Dit was eigenlijk de grootvader van C want uit BCPL werd de taal B ontwikkeld.

Met B en Assembler zou door Ken Thompson het Unix bedrijfssysteem ontwikkeld worden, echter B had nog enige tekortkomingen dus moest er een alternatief komen. Dit werd geboden door Dennis Ritchie. Ritchie had een taal ontworpen op een PDP-11 in 1972. Voor alle duidelijkheid, de PDP-11 is een computer. Er bestonden toen nog geen computers met een naam als Amiga. Nu weer even terug waar we waren gebleven, de taal die Ritchie had ontworpen was C. Met C werd toen het UNIX1 besturingssysteem ontwikkeld. Overigens, voor diegenen die het nog niet wisten, ook de eerste versie van het Amiga besturingssysteem werd met de taal C geschreven. In 1978 werd de definitie van C opgesteld in de appendix 'C reference manual', van het boek 'The C programming Language-Reference Manual'. De recente standaard werd in juni 1983 opgesteld door Dennis Ritchie. Tot zover deze historische inleiding.

Ferrari tegen Mercedes

Waarom C zult u zich afvragen. BASIC is toch ook goed genoeg? Dat zullen wij niet ontkennen, maar stelt u zich eens voor. U rijdt op de snelweg in uw net aangekochte Mercedes 500 SEC. U ziet de snelheidsmeter voor u gestaag klimmen naar 200 km/p uur. (Oeps, sorry minister Korthals-Altes) U kijkt argeloos in uw achteruitkijkspiegel. Ahhh, denkt u, niets geen verkeer op de weg, lekker rustig. U start de nieuwste CD van Matt Bianco. Hmm, heerlijk, u draait de volume knop op 10. Blik op oneindig, oren in de speakers en het verstand op NULL. Even denkt u nog, wat een vreemde storing, is m'n CD-speler soms naar de kloten. VRRRRAAAOE-EEEMMMMM. Jaagt daar me toch even een Ferrari Testarossa voorbij. Dit is zo'n beetje een grove aanduiding van het snelheidsverschil tussen C en BASIC.

Na BASIC komt C

BASIC met al z'n degelijkheid is een zeer mooie taal om in te programmeren, maar voor diegenen onder ons die toch net dat beetje lef en moed hebben is de taal C bedoelt. C heeft minder gereserveerde sleutelwoorden dan menig andere taal. Ze werkt veelvuldig met pointers, libraries, etc. En nog last but zeker niet the least, C is een echte compiler taal. Dat wil zeggen, de programma's die u zelf schrijft moeten gecompiled worden naar machine code en vervolgens gelinked zodat u de programma's vanuit DOS op kunt starten. Zo, om nu de vervente BASIC programmeurs onder ons toch nog een kans te geven over te stappen naar het kamp van de moedigen, gaan we als volgt te werk. We beginnen met een serie basis keywords zowel in BASIC als in C uit te leggen zodat u niet onmiddelijk een geheel andere programmeer gedachtenwijze aan hoeft te leren. U zult echter na verloop van tijd merken dat dat wel nodig zal zijn maar zover zijn we nu nog niet.

BASIC

We beginnen eenvoudig. U zult het zich ongetwijfeld nog wel herinneren. Eén van de eerste dingen die waarschijnlijk ingetypt heeft is; PRINT

"Mijn naam is ....." Dit zullen we in C ook proberen.

```
main()
{
  printf("Mijn naam is .....\n");
}
```

Op de plaats van de punten (....) kunt u uw eigen naam invullen. Op het eerste gezicht lijkt het wel wat ingewikkelder dan een simpele PRINT opdracht, echter de eigenlijke printopdracht is PRINTF(). Waarom dan dat MAIN() ervoor. Wel, in de taal C bestaat een programma uit:

° definities van variabelen, constanten, structures.

° een aantal functies waarin het eigenlijke programma is verwerkt.

° zogenaamde preprocessor faciliteiten die niet in het eigenlijke programma worden gebruikt, maar voor de compiler van belang zijn.

Dus, u raadt het mischien al, MAIN() is de functie waarin het hoofdprogramma is verwerkt. Een C programma heeft altijd een MAIN() functie nodig. Vervolgens zijn er dan de accolades { en }. Ze dienen om aan te geven waar een functie-blok of een compound statement begint en eindigt. Een compound statement is een blok van opdrachten die uitgevoerd wordt als de conditie, die er voor staat, waar is. Bijvoorbeeld een 'IF' opdracht. Tenslotte is er dan nog de PRINTF() functie in het programmavoorbeeld opgenomen. PRINTF() voert de in de string staande tekens uit naar het beeldscherm. In deze string mogen een aantal stuurcodes staan. In BASIC kunnen we dit vergelijken met de CTRL-codes. Nog even terug naar het voorbeeld. In de PRINTF() functie staat '\n'. In BASIC geschreven zou de opdracht luiden: PRINT "Mijn naam is ."+CHR$(10)

Dit is echter niet nodig want BASIC zet automatisch al een Newline achter de PRINT opdracht. In C gebeurt dit niet als u dit niet vermeldt. Hiervoor zijn meerdere stuurcodes ontwikkeld. Voor de volledigheid noemen we ze even op.

Nul l	\0	**CHR$(0)**
Backspace	\b	**CHR$(8)**
Tab	\t	**CHR$(9)**
Newline	\n	**CHR$(10)**
Formfeed	\f	**CHR$(12)**
Carriage Return	\r	**CHR$(13)**
Double Quote	\"	**CHR$(34)**
Single Quote	\'	**CHR$(39)**
Backslash	\\	**CHR$(92)**

U mag de codes ook opgeven als een 1, 2 of 3-cijferige octale code. Octale code is het cijferstelsel dat opgebouwd is rond het cijfer 8. Neem bijvoorbeeld het cijfer 7. Hexadecimaal is dat 0x7 en octaal is dit 7. Bij 8 verandert er iets. Hexadecimaal blijft nog hetzelfde als decimaal 0x8, echter octaal wordt 10. Stel dat u dan een Newline op wilt geven als 3-cijferige octale code, dan dient u in te typen, '\012' immers 8 decimaal is 10 octaal, 9 is 11 en 10 is 12 octaal. Als laatste dan nog de ';' achter de PRINTF() functie. Dit teken dient voor de compiler als einde statement opdracht. Zou u dit vergeten, dan zou de compiler niet doorhebben dat de opdracht afgelopen is. Het is namelijk mogelijk het programma te schrijven als een lees-



bare tekst. Een commando kan dan opgesplitst worden in 2 delen waardoor het geheel beter leesbaar wordt.

C en variabelen

BASIC maakt het u wat betreft het gebruik maken van variabelen wel heel erg gemakkelijk. Gewoon in het programma gebruiken is al voldoende. Helaas, dit grapje gaat echt niet op in C. U zult netjes voor het begin van het programma de computer moeten vertellen wat voor variabelen u wilt gebruiken zodat hiervoor ruimte in het geheugen vrijgemaakt kan worden. Nu komen we een beetje op glad ijs aangaande de verschillende variabelen typen. Niet elke BASIC variant gebruikt dezelfde manier van variabelen declaratie. We zullen hierom maar zoveel mogelijk naar AmigaBasic verwijzen. Eerst maar weer een voorbeeldje. Het BASIC variant:

```
leeftijd% = 18
naam$ = "Jan"
PRINT"Hallo ";naam$;
PRINT"dus je bent ";leeftijd%;" jaar."
```

Het zal wel duidelijk zijn wat de uitvoer hiervan is. Nu de C versie van het BASIC programma.

```
main()
 {
    int leeftijd;
    char *naam;
      naam = "Jan";
    leeftijd = 18;
      printf("Hallo %s dus je bent %2d jaar.  \n",naam,leeftijd);
 }
```

Zoals u ziet, moeten eerst een aantal variabelen opgegeven worden. In dit geval zijn dat leeftijd en naam. De variabele leeftijd is van het type INT en naam is van het type CHAR. Nu staat voor 'naam' nog een sterretje. De werkelijke betekenis daarvan wordt in een van de komende afleveringen behandeld. Neemt u voorlopig maar aan dat dit nodig is om een string op het beeldscherm te printen. Vervolgens worden de waarden aan de variabelen toegekend. Als laatste wordt het geheel afgeprint. Nu ziet u de werkelijke kracht van de PRINTF() functie. U kunt de uitvoer juist geformatteerd op het beeldscherm zetten. Voor deze geformatteerde uitvoer zijn een aantal conversie tekens beschikbaar. Dit zijn:

%c - uitvoer als een teken
%d - als een decimale integer
%e - als een getal met drijvende komma en wetenschappelijk notatie
%f - als een getal met drijvende komma
%g - in het %e of %f formaat, welke maar het kortst is
%s - uitvoer als een string.

Nu kunt u voor de getalsconversie-codes een getal vermelden. Dit getal stelt dan de format voor. Bijvoorbeeld; '%2d' betekent dat er ruimte voor een 2-cijferige integer waarde in de tekst vrijgemaakt moet worden. Bij floating point conversie tekens is dit nog even iets anders. Een voorbeeld: '%2.2f' wil zeggen dat er 2 getallen voor de komma en 2 getallen achter de komma staan bij het floating point getal. Tot zover de conversietekens. We hebben het al gehad over verschillende datatypen. Bij AmigaBasic is dit achter

een variabele aan te geven door middel van een bepaald teken of met behulp van het DEF commando. Zo kent AmigaBasic de volgende variabele typen; SHORT INTEGER, LONG INTEGER, SINGLE PRECISION, DOUBLE PRECISION en STRING . C kent identieke typen variabelen en daarnaast nog een aantal andere. We zullen ze de volgende keer doornemen.

De invoer

Net zoals in BASIC is het ook in C mogelijk variabelen tijdens de programmaloop een waarde te geven. BASIC heeft hiervoor de bevelen INPUT of INKEY$. C heeft hiervoor de functies SCANF() en GETCHAR(). SCANF() laat zich vergelijken met de BASIC equivalent INPUT. Het programma wacht met de uitvoer tot er een carriage return geweest is. Dit in tegenstelling tot GETCHAR(). GETCHAR() wacht niet tot er een carriage return geweest is. Daarmee is GETCHAR() dus te vergelijken met INKEY$. Een voorbeeld:

```
main()
  {
  char naam;
  printf("Wat is uw naam :");:
scanf("%s",naam);
    printf("Hallo %s. hoe gaat
het?\n",
naam);
    }
```

Ook hiervan is het BASIC equivalent beschikbaar.

```
INPUT "Wat is uw naam";naam$
 PRINT "Hallo ";naam$;" hoe gaat
het?"
```

Zoals u al bij de C-versie kon zien, ook SCANF() kent de conversietekens. Ze zijn vrijwel gelijk aan de conversietekens bij de PRINTF() functie. Er zijn wel enkele verschillen. Deze zijn; %e en %g ontbreken. %lf is er bijgekomen. %lf betekent, vertaal naar een getal met drijvende komma, dat wil zeggen naar een DOUBLE. Zoals we al gezegd hebben haalt de functie GETCHAR() een teken van het toetsenbord zonder te wachten op een Carriage Return. Aan de hand van het volgende voorbeeld zal het u duidelijker worden wat het precies doet.

```
main()
  {
      char c;
          while(1) { c = getchar();
            putchar(); }
  }
```

Meteen hier achteraan het BASIC equivalent zodat u de overeenkomsten en verschillen kunt zien.

```
WHILE 1 c$=INKEY$
 print c$;
WEND
```

Nu ziet u meteen wat PUTCHAR() en WHILE() uitvoeren. Het valt wel een beetje buiten het bestek van dit artikel, echter u kunt zelf wel uitvinden wat ze betekenen door naar de programma's te kijken. Als u er niet achter komt, geen paniek, in een volgend artikel worden ze wel behandeld. In het voorbeeld worden met behulp van GETCHAR() de tekens van het toetsenbord afgehaald en met behulp van PUTCHAR() op het beeldscherm gezet. Doordat de conditie altijd waar is in de WHILE-lus zal het programma in een eindeloze lus blijven hangen. Wij hopen dat tegen de tijd dat de volgende Commodore INFO uit komt u er klaar voor bent.

JOHAN & JOHAN

Alweer de derde aflevering van deze CLI cursus. 'How time flies when life goes fast!!'. In deze aflevering gaan we in op de zogenaamde 'startup-sequence' en alle commanddo's die in de startup-sequence (kunnen) worden gebruikt. Tevens zullen we wat dieper op ED in gaan.

Binnenin AmigaDOS 3

De startup-sequence

Wat is deze, wellicht al vaak gehoorde kreet, 'startup-sequence' nou eigenlijk? Het woord zegt het al, de opstart. Nadat de Amiga de zogenaamde 'bootprocedures' heeft doorgemaakt, dit wil zeggen, nadat ze het AmigaDOS heeft ingeladen, zal ze zoeken naar een file genaamd 'startup-sequence'. Deze file moet aanwezig zijn in de 's'-directory anders wordt deze opstartprocedure overgeslagen en komt de Amiga gewoon terug in het AmigaDOS.



Stel, u heeft een Amiga 500 met een geheugenuitbreiding. Hierop is een klok aanwezig. Als u dan automatisch de goede tijd ingelezen wilt hebben zou u een startup-sequence aan kunnen maken die deze functie vervult.
Hoe maakt u een dergelijke startup-sequence. Dat is niet zo moeilijk. U neemt een editor. Dit kunnen verschillende zijn. Wij behandelen er echter maar één. Dat is de editor die standaard op de Workbench diskette staat, genaamd ED.

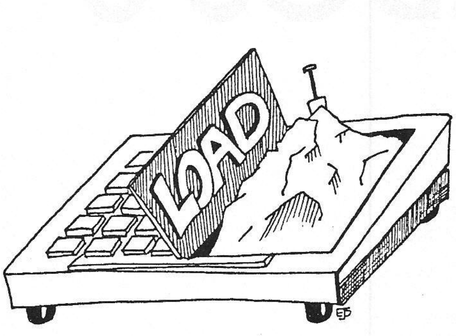
ED

ED is een doodgewone screeneditor. Niet al te bijzonder maar omdat iedereen over deze editor beschikt is deze toch van groot belang. De syntax van ED luidt: ED (programmanaam) (SIZE).
Nadat een programmanaam opgegeven is, wordt een venster geopend. Als de file ook echt bestaat zal aangegeven worden dat de file geladen wordt. Het aantal '@' tekentjes geeft globaal aan hoe lang de file is. Als de file niet bestaat zal er ook een venster geopend worden met daarin de vermelding 'Creating new file'. ED ondersteund de 1 optie. Dit is de optie **SIZE**. Achter SIZE kunt u vermelden wat de grootte moet zijn van het werkgeheugen. Bijvoorbeeld*: ED daphne.c SIZE 80000* Dit zal een bestand inlezen met de naam 'daphne.c' en zal een werkgeheugen reserveren voor dit bestand van 80000 bytes.
Nog een laatste opmerking. Het is niet mogelijk met ED programma files in te lezen. ED zal dan wel een venster openen maar ze zal vermelden; 'file contains binary'. Dat wil zeggen, de file bestaat niet uit ASCII data. ED kan niets anders lezen dan ASCII en zal dus het bestand niet inlezen.

De opdrachten van ED

ED kent twee manieren van opdrachten invoeren. 1. De CTRL-toets + een andere toets. 2. De ESC toets.
De CTRL toets combinatie wordt ook wel de directe mode genoemd. U drukt de CTRL toets in, tegelijkertijd drukt u een andere toets. We zullen alle combinaties eens doornemen. CTRL-A: Voeg een regel tussen. CTRL-B: Wis een regel. CTRL-D: Scroll'd de tekst omlaag. CTRL-E: Toggle, ga naar top of bottom van het scherm. CTRL-F: Toggle, draait hoofdletters om naar kleine letter en vica versa. CTRL-G: Herhaal de laatste ESC letter combinatie. CTRL-H: Zelfde als BACKSPACE. CTRL-I: Idem als TAB. CTRL-M: Gelijk aan RETURN. CTRL-O: Hetzelfe als DELETE. CTRL-R: Cursor naar vorig woord. CTRL-T: Cursor naar volgend woord. CTRL-U: Scroll'd de tekst om-

hoog. CTRL-V: Beschrijft opnieuw het scherm met dezelfde tekst (verify). CTRL-Y: Wist tekst tot het einde van de regel. CTRL-(vierkant haakje openen):Zelfde als ESC. CTRL-(vierkant haakje sluiten):Ga naar het einde van de regel. Deze commando's kunnen ingegeven worden tijdens het werken aan de tekst cq. het programma.



Als tweede zijn er dan de ESC commando's. Dit zijn 1- of meerletterige commando's die voorafgegaan worden door een ESC teken. Dit ESC teken wordt voorgesteld door een '*'. Ook hiervan is weer een hele waslijst. We behandelen ze in alfabetische volgorde. ESC A "tekst": hiermee voegt u de tekst, die tussen aanhalingstekens, staat in uw tekst of programma in. ESC B : ga naar het einde van het document/programma.
ESC BE: einde van het blok op de positie van de cursor. ESC BF/string/: zoek vanaf de cursor locatie tot de top van het document/programma naar de tussen '/' opgegeven string. ESC BS : zet de start van het blok op de cursor locatie. ESC CE: beweeg de cursor naar het einde van de regel.
ESC CL: beweeg de cursor 1 positie naar links. ESC CR: beweeg de cursor 1 positie naar rechts. ESC CS: beweeg de cursor naar het begin van de regel.
ESC D: wis de regel waarop de cursor staat. ESC DB : wis het blok dat u met ESC BS en ESC BE opgegeven hebt. ESC DC: wis het karakter waarover de cursor staat.
ESC E/string1/string2/: deze opdracht verwisselt 1 keer string1 voor string2. Als u ESC RP er voor vermeldt herhaalt het proces zich. ESC EQ/string1/string2/: hetzelfde als hierboven echter eerst wordt er geinformeerd of deze opdracht wel uitgevoerd moet worden. ESC EX: negeer in de regel waarin de cursor staat, de rechter kantlijn.
ESC F/string/: zoek vanaf de cursor locatie tot de onderkant van de file naar de tussen '/' vermelde string.
ESC I/string/: voeg de tussen '/' vermelde string tussen, voor de regel waarop de cursor staat. ESC IB : voeg een kopie van het blok, gedefinieerd met ESC BS en ESC BE, tussen achter de regel waarop de cursor staat. ESC IF/filenaam/: voeg de file genaamd 'filenaam' tussen, achter de regel waarop de cursor staat.
ESC J : voeg de regel, die onder de regel waarop de cursor staat, samen. De regel, die er aan toe wordt gevoegd, wordt achter de regel, waarop de cursor staat, gezet.
ESC LC: dit commando zorgt ervoor dat er WEL onderscheidt gemaakt wordt tussen hoofdletters en kleine letters tijdens zoekpogingen. Bijvoorbeeld: **lcs/Test/.* Dit zorgt er voor dat er gezocht wordt naar 'Test' en niet naar 'test'.
ESC M n : spring naar regel n. Voor n vult u een willekeurig getal in.
ESC N : spring naar het begin van de volgende regel.
ESC P: spring naar het begin van de voorgaande regel.
ESC Q: verlaat ED zonder te saven. Als u wijzigingen in de tekst/programma gemaakt heeft zal ED vermelden, 'Edits will be lost - type Y to confirm:'. Als u hier een ander karakter dan 'y' intypt zult u ED niet verlaten.
ESC RP: herhaal totdat er een fout optreedt. Bijvoorbeeld, **rpe/print/printf/.* Dit zorgt ervoor dat alle 'print' opdrachten in 'printf' veranderd worden. Het proces herhaalt zich totdat er een fout optreedt. Dit zal meestal de 'Search failed' error zijn.
ESC S: zet het gedeelte van de regel na de cursor locatie op de volgende regel. In feite is dit het zelfde als een RETURN geven op deze plaats. ESC SA: save de tekst/programma naar de file die u bij het opstarten opgegeven heeft. ED vermeldt dan 'Writing to file (filenaam)'. ESC SB: hiermee wordt het blok op het scherm afgebeeld. Echter alleen de blokstart (ESC BS) positie wordt hiermee duidelijk. Vanaf de top van het scherm wordt het blok dan afgebeeld. ESC SH: hiermee wordt status informatie omtrent uw file afgebeeld. Als eerste wordt de filenaam vermeld. Het tweede item is de TAB-afstand. Daarna worden de verschillende margins vermeld. Er wordt vervolgens afgebeeld wat voor tekst er op de blok start en blok einde regel staat, en als laatste wordt nog vermeld hoeveel van de van te voren opgegeven bufferruimte ingenomen is. ESC SL n: de linker kantlijn laat zich met deze functie instellen. ESC SR n: idem als voorgaande, alleen hiermee is de rechter kantlijn in te stellen. ESC ST n: hiermee stelt u de TAB afstand in.
ESC T: ga naar de top van uw tekst/programma.
ESC U: als u veranderingen gemaakt heeft en het is u toe niet geheel bevallen dan kunt u met ESC U de veranderingen weer ongedaan maken. Voorwaarde is wel dat u niet van de regel afgeweest bent. ESC UC: maak geen onderscheid tussen hoofd- en kleine letters tijdens zoekpogingen.
ESC WB/filenaam/: schrijf het blok, dat u met ESC BS en ESC BE gedefinieerd heeft, weg naar de file met de naam 'filenaam'.
ESC X: verlaat het programma maar schrijf eerst de gegevens weg naar de file die u van te voren opgegeven heeft. Nu we de commando's van ED een beetje kennen is het mogelijk een startup-sequence samen te stellen.

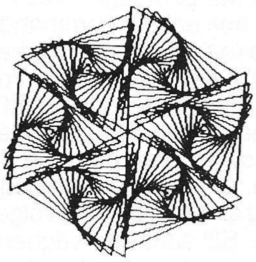


Het samenstellen

We beginnen door in te typen: format DRIVE (drivenummer) NAME testje Nu zal het even duren voordat uw schijf geformatteerd is. Als dit gebeurd is typt u: install (drivenummer). Tot zover het opstartbaar maken van de schijf. Voor een startup-sequence is ook nog een 's'-directory nodig. makedir (drivenummer):s zorgt hier- voor. Nu wordt het tijd voor de startup-sequence. Het volgende wat u in gaat typen is al een heel stuk op de goede weg. ed s/startup-sequence. We komen nu in ED terecht. Nu kunnen we een startup-sequence samen stellen. Deze zou er bijvoorbeeld als volgt uit kunnen zien.

```
echo "*ec" prompt A s/virusdetect
echo "Tracing system for
      virusactivity.."
echo "if any value comes up..
      BEWARE !!."
run s/virussearch path df2:
add dir ram: path ram:
add addbuffers df0: 20 addbuffers
      df2: 20 loadwb -debug;
```

Alleen het commando 'echo' is nog niet behandeld, dus wat de functie is van de overige commando's weet u ondertussen. 's/virusdetect' en 's/virussearch' zijn kleine machinetaal programmaatjes die controleren op de aanwezigheid van virussen op schijf en in het geheugen. Ze zijn niet van belang in deze cursus en worden daarom ook niet doorgenomen. We gaan nu eerst een aantal commando's doornemen die van belang zijn bij het opstellen van een startup-sequence.



Echo

Met ECHO kunt u een tekenreeks uitvoeren naar het beeldscherm. De syntax van ECHO is: ECHO "tekenreeks". Bijvoorbeeld. *ECHO "Hallo luitjes."* zal een vrij simpele tekst uit voeren naar het beeldscherm. Er zal komen te staan: 'Hallo luitjes.' ECHO kent enige opties. Deze opties worden ingegeven in de tekststring. De opties zijn:

*ec : maak het beeldscherm schoon
*n : geeft een NEWLINE. Mischien dat er nog meer opties zijn, na 832 verschillende opties uitgeprobeerd te hebben werdt het ons enigzins zwart voor de ogen. Ook ECHO kent weer de redirection mogelijkheid. ECHO ptr: "Hallo" stuurt de tekst 'Hallo' naar de printer.

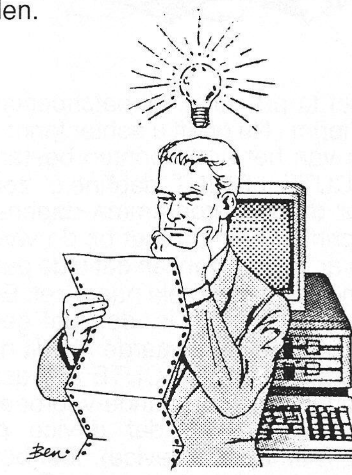
FAILAT Elke CLI instructie levert een nummer op als er iets mis gaat bij de uitvoering van het commando. AmigaDOS breekt de uitvoering van een batchbestand af als de foutcode groter is dan 10. Als u nu bijvoorbeeld een niet uitvoerbaar bestand uit probeert te voeren zal dit de foutcode 121, 'file is not a object module' opleveren. Doormiddel van FAILAT kunt u de fout- drempel, waarbij een batchbestand wordt afgebroken, verhogen of verlagen. De syntax van FAILAT. FAILAT n. Hierbij is n een integer waarde tussen 1 en en een getal met 17 nullen. Dit laatste hoeft u niet te proberen. De AmigaDOS foutmeldingen gaan maar tot 226, dus een getal even groter dan 226 is voldoende om uw batchbestand zonder onderbrekingen van AmigaDOS uit te laten voeren.

SKIP/LAB Een soort GOTO commando in AmigaDOS? Tja, dat is wel ongeveer de vergelijking die je kunt trekken. Als AmigaDOS SKIP tegenkomt zoekt het in het bestand naar het woord LAB en voert daarvandaan het bestand verder uit. Achter SKIP en LAB mag ook een labelnaam vermeld staan. Bijvoorbeeld, *IF NOT EXIST daphne.c SKIP huppeldepup ENDIF TYPE daphne.c*

LAB huppeldepup (de rest van het bestand) Ok, we geven toe een beetje een onlogische constructie, maar ja, het voorbeeld heiligt de middelen nietwaar? Toch is het niet aan te raden hiervan vaak gebruik te maken. Anders wordt het net een BASIC programma. Onoverzichtelijk dus.

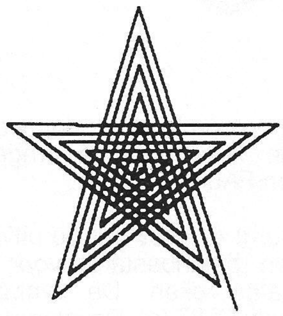
WAIT Het commando zegt het al. Wachten. Omdat de Amiga een multitasking systeem is, kan het voorkomen dat meerdere taken bijvoorbeeld de diskdrive aan willen spreken. Om nu te voorkomen dat u langer zult moeten wachten dan als u de taken na elkaar uitvoert is het mogelijk de WAIT instructie te gebruiken. De syntax luidt: WAIT (UNTIL) n (eenheid).

'n' is of een integer waarde, of een tijdstip. Bijvoorbeeld. WAIT 2 sec. Er wordt nu 2 seconden gewacht, echter er mag ook '2 min' ingevuld worden. De task zal dan 2 minuten wachten. Ook is het met behulp van het keyword 'UNTIL' mogelijk de task tot een bepaald tijdstip te laten wachten. WAIT UNTIL 12:35 zorgt er voor dat de task tot 12:35 wacht. Als u onderwijl toch schoon genoeg krijgt van de TE lang ingestelde wachttijd, een simpele druk op de CTRL-C combinatie verlost te task uit zijn WACHTELIJK lijden.



IF/ELSE/ENDIF. Hiermee wordt het mogelijk aan de hand van bepaalde voorwaarden een gedeelte van het programma wel of niet uit te voeren. De drie instructies IF/ELSE/ENDIF behandelen we in 1 keer omdat ze onlosmakelijk met elkaar zijn verbonden. De een mag niet los van de ander voorkomen. De syntax: IF (conditie) opdrachten (ELSE opdrachten) ENDIF Het wordt waarschijnlijk al duidelijk. Er moet na het IF commando een conditie komen. Aan de hand van de resulterende waarde (TRUE or FALSE) wordt een bepaalde programmastroom gevolgd.

Er bestaan een zestal condities. Deze zijn: -**EXISTS** : hierbij wordt op schijf naar het achter EXISTS vermelde bestand gezocht. Als het bestaat is de uitkomst TRUE en als het niet bestaat is de uitkomst FALSE.

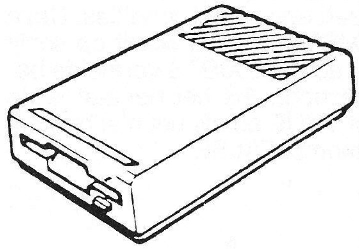


-EQ: equal, engels voor gelijk aan. Er wordt getest of twee strings gelijk aan elkaar zijn. Als de strings gelijk aan elkaar zijn is de uitkomst weer TRUE anders is het FALSE. Een voorbeeld: IF "is dat zo" eq "is dat niet zo" ECHO "ja hoor" ELSE ECHO "nee niet zo" ENDIF Dit levert als uitkomst FALSE op. Dus zal het ECHO statement tussen IF en ELSE niet uitgevoerd worden, echter het ECHO statement achter ELSE zal wel uitgevoerd worden. Immers de uitkomst was FALSE. Echter twee tekststrings met elkaar vergelijken is onzin. Er is wel een betere toepassing. Met de .key en .def instructies in een batchbestand zijn betere constructies denkbaar. Hier komen we straks nog op terug.

-FAIL: met deze conditie is het mogelijk te controleren of de voorgaande opdracht een fout heeft veroorzaakt. Het foutnummer hiervan moet wel groter of gelijk zijn aan 20. Het is echter wel noodzakelijk dat u met de instructie FAILAT de foutdrempel boven de 20 zet, anders wordt het batchbestand nooit tot zover uitgevoerd, immers AmigaDOS stopt standaard met het uitvoeren van een batchbestand, bij een foutwaarde van 10.

-ERROR: hetzelfde als hierboven,

echter de foutdrempel ligt nu bij 10.
-WARN: ook weer soortgelijk, nu ligt de foutdrempel bij 5.
-NOT : als laatste de toevoeging NOT. NOT kan worden gezet voor de overige condities. Bijvoorbeeld, IF NOT EXISTS (filenaam) In feite wordt de redenering hierdoor omgedraait. Als de bovenstaande EXISTS een TRUE



waarde op zou leveren, dan wordt dit door de NOT conditie omgedraait naar een FALSE waarde.

QUIT zorgt er voor dat de uitvoering van een batchbestand voor- tijdig wordt afgebroken. De syntax van QUIT luidt: QUIT (n). De integer n achter QUIT geeft de mogelijkheid een return waarde mee te geven. Als dit groter dan 10 is, en u voert de instructie uit van uit CLI zal er op uw beeldscherm verschijnen; 'quit failed returncode (een code 10)'.
ASK Een commando dat het mogelijk maakt interactief te werken op de Amiga. Weliswaar op een simpel niveau, maar toch. In combinatie met IF is ASK een van de krachtigste CLI commando's. De syntax van ASK: ASK (string). Uw antwoord op ASK dient dan 'Yes,No,Y, N' te zijn. Als uw antwoord 'Yes of Y' is dan levert ASK een foutcode 5 op. Met behulp van IF WARN zou u kunnen testen op deze gebeurtenis. Het is ook mogelijk de foutdrempel te verlagen waardoor u weer terug in CLI kan komen. Een voorbeeldje. U wilt in uw startup- sequence de mogelijkheid kunnen krijgen om de WorkBench wel of niet in te laden. De startup-sequence zou er als volgt uit kunnen zien, ASK "Wilt u de WorkBench inladen? (y/n)" IF WARN LOADWB ENDIF
EXECUTE Als laatste het commando dat voor batchbestanden zo belangrijk is. Zonder EXECUTE zou het niet mogelijk zijn batchbestanden uit te voeren. Dus, EXECUTE dient om batchbestanden uit te voeren. Wat is nu een batchbestand? Een batchbestand is een verzameling van DOS opdrachten samengevoegd in 1 ASCII-file. Een voorbeeld van een batchbestand is bijvoorbeeld de startup-sequence. Om nu een batchbestand uit te voeren moet het EXECUTE commando ten tonele verschijnen. Typt u maar eens in: EXECUTE s/startup-sequence. De startup-sequence van uw AmigaDOS schijf zal uitgevoerd worden. Nog even de syntax van EXECUTE, EXECUTE (naam van het batchbestand). Wat heeft dit allemaal met de startup-sequence van doen zult u zich nu afvragen. Zoals we net al hebben gezegd, de startup-sequenc is een batchbestand dus al de hierboven behandelde zaken zijn van toepassing op de startup-sequence. Zoals we al eerder gezegd hebben in het IF/ELSE/ENDIF onderdeel, er zijn nog een paar extras met batchbestanden. Het is mogelijk om met variabelen te werken. Echter in tegenstelling tot de programmeertaal die de meesten van ons wel zullen beheersen, BASIC, moeten variabelen eerst gedeclareerd worden. Dit kan niet midden in het programma gebeuren of op meerdere plaatsen, nee het moet in het begin gebeuren en mag maar 1 keer gebeuren. Met de instructie .key declareert u de variabelen.
Een voorbeeldje: .key naam TYPE



(name) to prt: Stel, dit batchbestand heet 'lprint'. Nu geeft u achter lprint de naam van het uit te printen bestand. 'EXECUTE LPRINT daphne.c' zorgt ervoor dat het programma daphne.c uitgeprint wordt. U moet bij de variabelen er wel om denken dat u de punthaken om de variabele naam zet. Een volgende instructie is .def .def geeft een variabele een waarde als dit niet vermeld is in de EXECUTE opdracht. Nog even het voorgaande voorbeeld: .key naam,device .def device ptr: TYPE (name) TO (device). .def betekent dus de default waarde. Default zal in dit voorbeeld dus de uitvoer naar de printer gaan. We komen nu weer terug bij het uitgangspunt van deze aflevering, de startup-sequence. Met onze nieuwe kennis gewapend zullen we proberen een functionele startup-sequence te maken.

De uitgangspunten

Wat willen we in onze startup-sequence hebben. Er moeten in ieder geval buffers voor de verschillende diskdrives gecreerd worden, dit verhoogd immers de snelheid van het werken met de disk- drives. Tevens moet er gevraagd worden of de WorkBench in geladen moet worden of of er een RAM disk aangemaakt moet worden. We willen ook enkele commando's resident maken zodat het werken met AmigaDOS wat flexibeler wordt. Met deze uitgangspunten 'in mind' gaan we aan het werk. We beginnen met het veilig stellen van de oude startup-sequence. COPY s/startup-sequence s/startup zal dit doen. Vervolgens typen we in: ED s/startup-sequence.
Nu kunnen we beginnen met de startup-sequence. :

```
ADDBUFFERS df0: 20
IF EXISTS df1:
      ADDBUFFERS df1: 20
      PATH df1: ADD
ENDIF
IF EXISTS df2:
      ADDBUFFERS df2: 20
      PATH df2: ADD
ENDIF
ASK "Wilt u een ram disk aanmaken
      (yes/ no)?"
IF WARN
      COPY c/dir TO ram:
      COPY c/copy TO ram: CD ram:
      COPY df0:c/cd TO ram:
      COPY df0:c/echo TO ram:
      COPY df0:c/type TO ram:
      ECHO "dir, copy, cd, echo en type zijn
      nu resident"
ENDIF
ASK "Wilt u naar de WorkBench
      (yes/no)"
IF WARN
      LOADWB
      ENDCLI NIL:
ENDIF
```

Zo, met behulp van de voorgaande delen van deze cursus moet u zelf kunnen bepalen wat deze startup-sequence nu uitvoert. Onthoudt u wel, als u naar CLI toegaat, de default directory, dus de directory welke u opgeeft bij het CD commando, is nu de RAM disk. Dus als u een directory wilt hebben van df0: dient u in te typen, DIR df0: en niet DIR, want dan krijgt u de directory van de RAM disk.

Waarschijnlijk heeft u er al maanden geleden over gehoord, echter in Nederland is ze nog niet te verkrijgen. We hebben het over WorkBench v1.3. In Duitsland is de onofficiele versie al geruime tijd in het grijze circuit te verkrijgen, zodat velen onder ons al een copy van een copy van een copy etc. etc. zullen hebben. Toch zijn er weer een aantal verschillen met deze versie te vinden. In dit artikel een kort overzicht van deze WorkBench-update.

WORKBENCH V1.3

Nieuwe commando's voor AmigaDOS

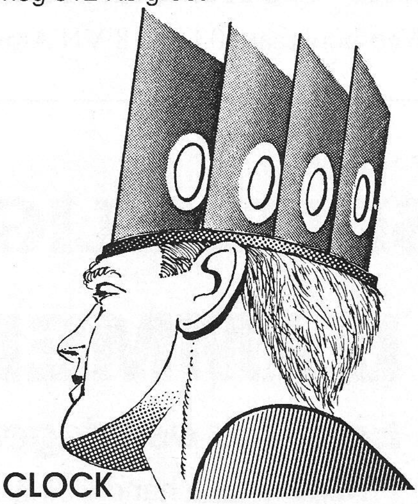
Met de komst van WorkBench v1.3 zijn er een aantal commando's toegevoegd aan de lijst van al bestaande AmigaDOS commando's. Wat voor nieuwe commando's zijn er nu gekomen. Als eerste een zeer nuttig commando. Waarom bestond het nog niet eerder!

Resident

Recht vertaald betekent het 'aanwezig'. Dit is wat het commando eigenlijk doet. Het zorgt ervoor dat AmigaDOS commando's permanent, zolang er geen reset gegeven is, in het geheugen staan. Er is wel een beperking. Alleen commando's waarvan het 'pure' bit gezet is kunnen resident gemaakt worden. Het 'pure' bit is 1 van de nieuwe status bits. Als u de CLI cursus gevolgd heeft, hebt u vast wel eens de statusbits 'rwed' gezien. Met de nieuwe WorkBench release zijn nog een aantal statusbits toegevoegd. Dit zijn 'H'idden, 'A'rchive en 'P'ure. Commando's waarvan het 'pure' bit niet gezet is zullen waarschijnlijk niet werken. RESIDENT werkt alleen maar in de nieuwe SHELL-omgeving. Op deze SHELL komen we straks nog terug.

Avail

Voor Aztec C bezitters zal het **AVAIL** commando wellicht bekend zijn. De nieuwe WorkBench beschikt ook over het commando. Met het commando krijgt u een overzicht van het totale Chip-memory en Fast-memory. Voor de zekerheid nog maar even, Chip memory is dat deel van het geheugen waarin de zogenaamde 'Custom chips', de Blitter en Copper etc., vrijelijk hun gang gaan. Vaak dwarsbomen ze hierbij acties van de microprocessor. In dit geval de MC68000. In Fast memory kunnen de 'Custom chips' nog niet komen dus hier kan de MC68000 zonder onderbrekingen zijn werk doen. Het Chip memory is nu nog 512 Kb groot.



CLOCK

Het was bij WB 1.2 al mogelijk een klok op het beeldscherm te krijgen. Dit was nog niet mogelijk van uit CLI. Het commando CLOCK vervult deze tekortkoming. Naar wens is een Analoge of Digitale klok in te stellen met variabele grootte. In deze zelfde catagorie valt ook het commando **CLOCKPTR**. Na uitvoering van dit commando moet u maar eens op het WorkBench scherm klikken, dus op het scherm achter het CLI window. En voila, wat gebeurt er, uw mousepointer verandert in een digitaal klokje. Als u helemaal naar linksboven beweegt zal het klokje minuten en seconden aangeven, als u dan langs de linkerrand naar beneden beweegt zult u zien dat de datum verschijnt. De (ex-)64 bezitters zullen zich het BASIC commando CMD nog wel herinneren. Even een geheugen opfrisser. Wat deed dit commando ook al weer? Juist de uitvoer omleiden naar een randapparaat. Als u nu LIST intypte ging de uitvoer naar het opgegeven randapparaat. Goed, nu weer terug naar het eigenlijke verhaal, **CMD** in AmigaDOS dient om de uitvoer die naar het seriele of parallele device gaat, om te leiden naar een file. Bijvoorbeeld, voordat u de naar de printer wilt sturen maar u weet niet of alles wel goed op papier komt te staan. Als het dan ook nog uw laatste velletje papier is dan wilt u wel zeker weten of de uitvoer goed is. Goed, hier komt het commando CMD dus uitstekend van pas. Even intypen RUN CMD serial test.prt en de uitvoer die u daarna naar de seriele printer stuurt wordt omgeleid naar de file 'test.prt'. Probeer maar eens **COPY s/startup-sequence TO prt:**. Op het beeldscherm zal verschijnen: 'Redirected (aantal) bytes from serial.device to test.prt'. Nu kunt u in de file test.prt nakijken of de printer uitvoer wel correct geweest zou zijn.

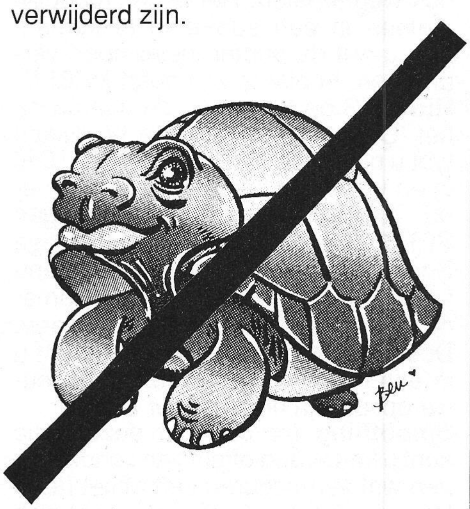
SKIP/ENDSKIP

Het SKIP commando bestond al maar nu is er het ENDSKIP commando aan toegevoegd. Achter het SKIP commando staat het label waar naar toe gesprongen moet worden. Als tijdens die sprong een ENDSKIP tegengekomen wordt zal de sprong afgebroken worden en wordt er doorgegaan met uitvoeren van de file. Het is ons al

meerdere keren overkomen dat we tijdens het werken in CLI voor een simpele rekenoperatie terug moesten grijpen op ons oude vertrouwde rekenmachine. Toch stond er dan een rekengigant voor ons. In CLI was het niet mogelijk een reken operatie uit te voeren. Dit is nu opgelost met het **EVAL** commando. EVAL kan simpele rekenoperaties uitvoeren. Het kan twee integer getallen optellen, aftrekken, vermenigvuldigen, delen, ANDen, ORen etc. etc.

FASTMEMFIRST

Dit is een DOS commando maar meer een utility. De bedoeling van dit commando is simpel, het probeert het Chip ram zo lang mogelijk vrij te houden. Programma's worden eerst in Fast ram geladen. **FF en FIXFONTS** zijn utilities die invloed hebben op de leesbare tekens op het beeldscherm. Met FF is de printsnelheid sneller te maken, tevens is het mogelijk om een andere karakterset in CLI te gebruiken. Met de optie -0 zet u FF aan, met -n zet u de snelle printroutine weer uit. FIXFONTS 'update' de .font files in de FONTS directory. Hiermee wordt bedoelt dat als u nu een font verwijderd, bijvoorbeeld u verwijderd de directory RUBY dan zal na een aanroep van FIXFONTS ook de file RUBY.FONT verwijderd zijn.



ICONX

Hiermee kunt eindelijk ook batchbestanden uitvoeren vanuit de WorkBench. Dezen waren anders alleen vanuit CLI met EXECUTE uit te voeren. U maakt een icon aan te voor het batch bestand en aanklikken maar. **INITPRINTER** forceert het printer.device de variabelen gedefinieert met de PREFERENCES in te lezen. Voor de harddisk bezitters die hun gegevens willen beveiligen is het commando **LOCK** handig. U kunt de write protect status van de drive of partitie zetten of resetten. Ook is het mogelijk een password mee te geven. De harddisk moet dan wel geformatteerd zijn met



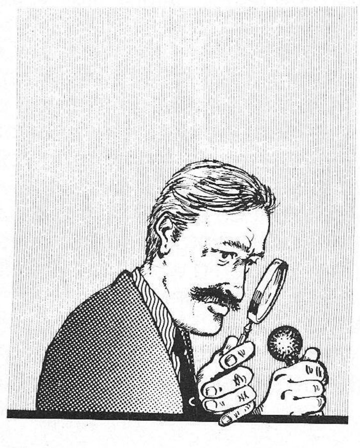
het zogenaamde FastFileSystem. **MORE**. Hoe vaak heeft u nu tijdens het TYPEn van een file niet op de spatiebalk zitten rammen. En hoe vaak heeft u na het TYPEn van de file niet de ellenlange rij 1 1 1 1.... gezien. Goed, de oplossing is dus MORE . MORE stopt de uitvoer van tekst aan het eind van elke bladzij. U drukt dan op de spatiebalk als u door wilt gaan of drukt CTRL-C of 'q' als u wilt beeindigen. MORE ondersteunt nog veel meer van dergelijke commando's. Hierover meer in een versie 1.3 update van de cursus 'Binnenin AmigaDOS'.

ALIAS

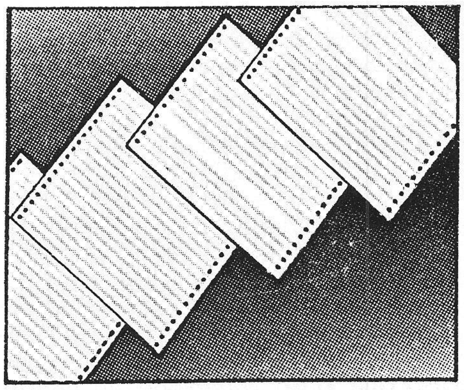
Weer een commando dat alleen werkt binnen de SHELL omgeving. ALIAS zorgt ervoor dat een bepaalde variabele een DOS functie krijgt. Bijvoorbeeld ALIAS schoon ECHO "*ec" Als u nu 'schoon' intypt zal het beeldscherm schoongemaakt worden. En dan als laatst, het commando **WHICH**. WHICH gaat na waar een bepaald commando is. Hij zoekt dus uit in welke directory het commando staat of dat het in de RESIDENT list staat. Hierbij zijn twee opties van belang. De NORES en de RES optie. De een zorgt ervoor dat er niet in de RESIDENT list gezocht wordt, de ander zorgt ervoor dat alleen in de RESIDENT list gezocht wordt.

Verbeterde commando's

AmigaDOS gaat steeds meer in de richting van MSDOS of UNIX. Op zich natuurlijk een zeer prettig en lovenswaardig streven. Veel bestaande commando's zijn verbeterd of uitgebreid. Om ze allen te bespreken is meer een taak voor een van de volgende delen van de 'Binnenin AmigaDOS' cursus.
Sommige commando's hebben echter grote veranderingen ondergaan. In alfabetische volgorde zijn dit:
CLOCK . Zoals al eerder gezegd, het commando bestond al in WB.1.2 maar ze is nu ook vanuit CLI op te starten met verschillende opties. Al bekende opties zullen zijn; ANALOG,DIGITAL. De optie DIGITAL2 zorgt ervoor dat er een window ter grootte van de 'titlebar', de bovenste rand van het window, geopend wordt. Hierin wordt dan de tijd afgebeeld.
COPY ondersteunt nu meerdere opties.
BUF bepaalt de buffergrootte voor het copieren.
CLONE zorgt ervoor dat alle statusbits meegekopieerd worden.
DATE zorgt ervoor dat de datum dat de file gecreeerd is meegecopieerd wordt.
DIR ondersteund nu ook de opties ALL, DIRS en INTER. Vroeger moesten dezen immers met OPT A, OPT D en OPT I meegegeven worden.
In de **INTER** mode kunt u na het vraagteken COM of C intypen. AmigaDOS vraagt vervolgens wat voor commando u uit wilt voeren. Zo kunt u dan een file TYPEn etcetera etcetera.
DISKDOCTOR controleert nu eerst de hoeveelheid vrije geheugen voordat het begint. Tevens werkt het nu samen met het nieuwe FastFileSystem.
ECHO . ECHO heeft een aantal nieuwe opties gekregen. NOLINE geeft aan dat er geen carriage return gegeven moet worden, FIRST en LEN bepalen wat er met de tekststring gebeuren moet. FIRST bepaalt van welke positie de tekst afgebeeld moet worden, LEN geeft de lengte van de afgebeelde string aan.
FORMAT . FORMAT kent ook een aantal nieuwe opties. Deze zijn; QUICK, FFS, NOFFS. QUICK is handig voor harddisk bezitters. Met QUICK wordt alleen de root block, de



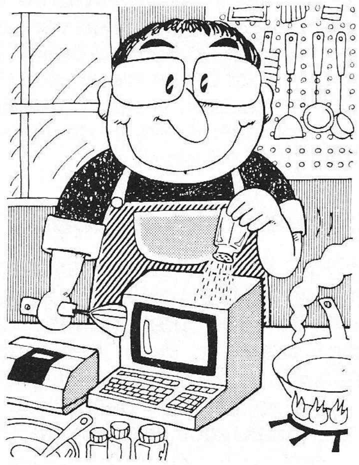
bootblock en de bitmap blocks opnieuw geformatteerd.
FFS betekent FastFileSystem. Hiermee geeft u aan dat u de harddisk of resetvaste ramdisk met het FastFileSystem wilt formatteren. NOFFS betekent, zoals er al staat, NIET FFS dus geen FastFileSystem gebruiken.
INSTALL . INSTALL kent een aantal verbeteringen ten aanzien van de virussen. Ten eerste wist ze het geheugen voordat het boot blok opgebouwd wordt. Ten tweede zijn er een aantal opties bijgekomen. CHECK bekijkt de bootsector en geeft hiervan verslag. De optie NOBOOT geeft aan dat de disk niet opstartbaar is maar dat het wel een DOS disk is.



LIST . Veel veranderingen (en dus veel uitleg) in het commando LIST. De nieuwe statusbits (hidden, archive en pure) worden nu door LIST afgebeeld. LIST kan ook met wildcards (#?) samenwerken. Ook kent ze een aantal extra opties, Deze zijn, FILES, DIRS en BLOCK. Bij de optie FILES worden alleen de files afgebeeld dus niet de directories, dit in tegenstelling tot DIRS waarbij alleen de directories afgebeeld worden. De optie BLOCK geeft aan dat de filegrootte in aantallen blokken moet worden afgebeeld. Als laatste en als mooiste, LIST kent een keyword waarmee de uitvoer geregeld kan worden, het is zelfs mogelijk hiermee kleine 'script' files te maken.
Een voorbeeldje: *LIST #? LFORMAT="protect %S -d"*. U ziet dus, met het LFORMAT keyword kan de uitvoer geregeld worden. De %S wordt vervangen door de filenaam. Binnen de string na LFORMAT kunnen zowel DOS commando's als ASCII tekst die afgeprint moet worden. In het bovenstaande voorbeeld wordt van alle files het delete bit gereset.
NEWCLI . NEWCLI heeft een opstart file genaamd CLI-STARTUP. Deze staat in de s-directory. In feite is er dus een default voor de al bestaande optie FROM.
PROMPT . Met PROMPT kunt u nu ook de actuele directory afbeelden. De optie '%s' zorgt hiervoor.
PROTECT . PROTECT werkt ook samen met de nieuwe protectie bits. Met de opties ADD of '+' en SUB of '-' kunt u bepalen of u een bit wilt zetten of resetten.
TYPE . TYPE kent ook een aanal nieuwe opties te weten, NUM en HEX. Ze bestonden vroeger ook al maar dan moest u OPT N of OPT H opgeven.

Nieuwe werkomgeving

Een nieuwe WorkBench, een nieuwe werkomgeving. Ze heet SHELL. Deze werkomgeving is veel flexibeler dan datgeen waar nu de meesten nog in werken. Veel van de nieuwe commando's werken alleen maar in deze SHELL.
Met het commando **NEWSHELL** start u een nieuwe SHELL op. Eerst wordt, net als met NEWCLI een startup-file opgestart. Deze heet SHELL-STARTUP en staat ook weer in de s-directory. Als u een shell opstart in WB1.3 zal de prompt niet met het bekende '1' terugkomen maar met '1.SYS:'. Dit is een van de verbeterde punten van deze shell, het vermeldt het actuele device en de actuele directory. Dit wordt gedaan door het verbeterde **PROMPT** commando. Immers met '%s' kan men de actuele directory of het actuele device laten verschijnen. Ook een hele verbetering is het feit dat het zogenaamde 'command line editing' mogelijk is. Dit betekent dat het mogelijk is met de cursor terug te gaan naar het begin van de regel om een eventuele fout te verbeteren. In CLI betekende dit automatisch weer de hele regel opnieuw intypen. Maar er is nog een extra. Doormiddel van de CURSOR UP & CURSOR DOWN toetsen kan men de vorige commandoregels weer oproepen.



Hierover meer straks bij het nog te behandelen NEWCON: device.
De al genoemde RESIDENT en ALIAS commando's hebben in de SHELL een functie. SHELL maakt gebruik van het zogenaamde 'script' bit. Het is dus niet meer noodzakelijk bij deze bestanden het EXECUTE commando te gebruiken, SHELL herkent aan dit script bit dat het bestand een batchbestand is en dat het dit bestand overeenkomstig moet behandelen.



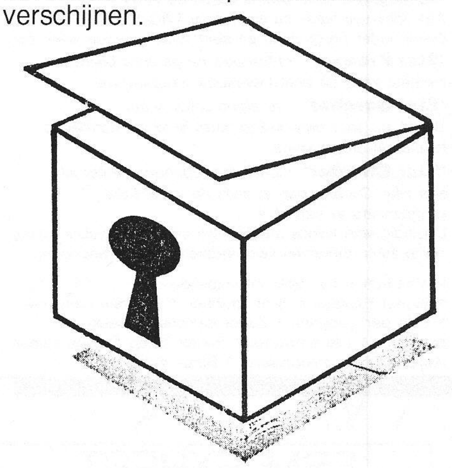
De preferences

Ook de preferences zijn een kleine wijziging ondergaan. De CLI ON/OFF optie is verdwenen. Dit was al vrij onzinnig dus reden te meer om dit dan ook weg te laten. Het is ook mogelijk meteen in een submenu te komen. Stel u wilt de printer instellingen veranderen. In plaats van eerst PREFERENCES op te roepen en daarna op het 'Change Printer' vakje te klikken typt u nu 'PREFERENCES PRINTER' in en u komt in het gewenste submenu. Zo komt u met de toevoegingen SERIAL en POINTER in het 'Change Serial' en het 'Edit Pointer' submenu terecht. Het 'Change Printer' submenu heeft een toevoeging gekregen. Door op 'Graphic 2' te klikken komt u in een nieuw submenu terecht. Nieuwe opties zijn hierin onder andere;
Smoothing (on/off) Met deze optie kunt u kartelrand effect van schuine lijnen wat verminderen Left offset (aantal inches) De linker kantlijn laat zich hiermee naar believen instellen. Center (on/off) Hiermee is het mogelijk het plaatje in het midden van het papier te zetten.
Limits De verschillende grenzen zijn hier mee in te stellen. Dit kan in inches of in pixels opgegeven worden. Tevens is een vermenigvuldiging factor hier instelbaar.
Color Correct Voor bezitters van kleurenprinters zeer handig. Hiermee is het mogelijk de kleuren op het papier wat feller te laten lijken dan op het

beeldscherm.

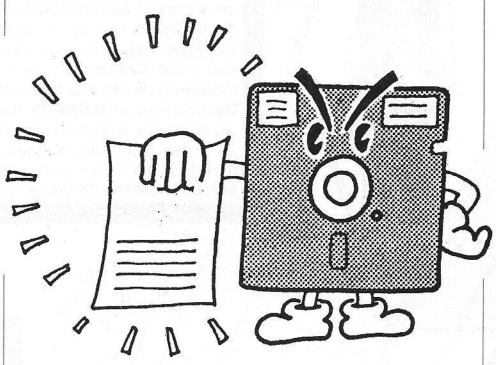
Devices en Handlers

Als je de L directory van WB 1.3 voor de eerste keer 'list' lijkt het wel of deze twee keer zo groot geworden is. Dit is mischien ook wel zo want er zijn welgeteld 5 nieuwe handlers bijgekomen en in de DEVS directory is er 1 nieuw device bijgekomen. De nieuwe handlers op een rijtje;
AUX-HANDLER. Een nieuwe seriele handler waarmee het nu ook mogelijk is geworden om met een terminal aan de Amiga te 'hangen'. Het gaat hierbij om ongebufferde seriele in- en uitvoer.
SPEAK-HANDLER . De uitvoer van uw net ingetypte tekst wilt u laten spreken? Dit kan. Typt u maar eens in; COPY w tekst bestand TO SPEAK: Met deze handler is dus mogelijk bestanden uit te laten spreken, letterlijk en figuurlijk.
PIPE-HANDLER . Tot nu toe was communicatie tussen twee tasks een zeer lastige zaak. Het was mogelijk maar niet simpel. Met de PIPE-HANDLER is het wat eenvoudiger geworden. COPY tekst.tst TO PIPE: zorgt ervoor dat het bestand tekst.tst gebufferd wordt door de pipe-handler. Als u nu met een ander programma of in een andere task *COPY pipe: tekst.tst TO ** doet dan zal de tekst van het bestand weer op het beeldscherm verschijnen.



NEWCON-HANDLER Deze is zo net bij de SHELL al even ter sprake gekomen. Met deze handler is het zogenaamde command line editing een feit op de Amiga. De NEWCON-HANDLER heeft een buffer van 2K waarin het vorige commando's opslaat. Als de buffer vol is zal het eerst ingetypte commando verwijderd worden. Men noemt zo'n buffer dan ook wel een 'circular buffer'. De controle karakters worden door de NEWCON-HANDLER in reverse weergegeven.

ENV-HANDLER . Dit is eigenlijk nog geen handler maar er wordt wel aan gewerkt. De gegevens worden nu nog in de directory ram:env weggeschreven.
Met deze handler en de commando's GETENV en SETENV is het mogelijk in DOS variabelen een stringwaarde te geven. Bijvoorbeeld, *SETENV demo "Deze tekst wordt zometeen afgedrukt"* zorgt ervoor dat aan de variabele 'demo' de daarachter getypte sting wordt verbonden. Als u nu weer intypt GETENV demo zal de ingetypte string weer afgebeeld worden.



Door het **RAMDRIVE-DEVICE** zal straks elke Amiga bezitter over een resetvaste ramdisk beschikken. Nu zult u zich afvragen, is het niet een beetje onnozel, twee ramdisks in het geheugen. Op zich wel, maar Commodore antwoordt hierop dat de niet resetvaste ram disk voor tijdelijke variabelen is en de resetvaste ram disk voor de programma's die men wil bewaren gedurende de tijd dat de computer aanstaat. Afdoende verklaring lijkt ons.

TURBO DRIVES door FFS

FFS. Drie letters die harddisk bezitters als muziek in de oren klinkt. Eindelijk nog meer snelheid uit m'n harddisk horen we ze al denken. FFS staat voor FastFileSystem. Voor harddisks bleek al snel het normale filehandle systeem van de Amiga te traag te zijn. Reden hiervan is dat het oude formaat eerst 24 bytes aan header informatie wegschrijft en vervolgens pas het data blok wegschrijft. Bij het inlezen is het proces precies andersom. Eerst moet bij het inlezen van elk blok deze 24 bytes er 'afgestroopt' worden. Dit kost ontzettend veel tijd als er veel blokken ingelezen moeten worden, bovendien waren de routines niet in 100% machinecode geschreven. Deze tijd is nu echt voorbij. Harddisk bezitters kunnen weer rustig ademhalen. Formatteer uw harddisk met het FastFileSystem en u heeft geen last meer van het oude file systeem. Het oude filesysteem is natuurlijk nog steeds te gebruiken. Het nieuwe filehandle systeem biedt nog een voordeel. U krijgt 4 procent meer ruimte op uw harddisk. Op een 20 MB harddisk is dit ongeveer 800 K, dus de moeite wel waard.

Extra's

Bij ons exemplaar van WB 1.3 zat ook weer een extras schijf. Alhoewel hierop geen nieuwe AmigaBasic versie aanwezig was vonden toch nog enkele kleine toevoegingen; -een aantal nieuwe fonts, te weten, Times, Helvetica en Courier. -een nieuwe MicroEmacs versie. -meerdere printerdrivers waaronder die voor de Brother HR-15XL en Canon PJ-1080A.

Conclusie

De nieuwe WorkBench, ook al is de officiele versie nog niet uit, is een hele verbetering ten opzichte van WorkBench 1.2. De echte verbeteringen zijn de nieuwe SHELL en het FastFileSystem. Toch blijven er kleine punten van kritiek zoals het afbeelden van de volume naam van een schijf in plaats van het afbeelden van de device naam maar dit zal waarschijnlijk een kwestie van smaak zijn. Toch is deze WorkBench een goede stap vooruit en zal waarschijnlijk bij de mensen die vaak in CLI werkten in goede aarde vallen.



Dan blijft er nog de kwestie van de nieuwe KickStart. Alhoewel Commodore zegt dat de v1.2 KickStart prima samen werkt met de v1.3 WorkBench blijft er toch wel het gevoel dat bij een nieuwe WorkBench een nieuwe KickStart hoort. Helemaal als u al over een HardDisk beschikt is dit zonder meer nodig. Geen gedoe meer met die schijven, dat is toch ideaal! Altijd opstarten van de harde schijf.
Dit belooft waarschijnlijk voor de computer reperatiebedrijven in ons land binnenkort weer drukke tijden. Er zijn per slot van rekening nog veel Amiga bezitters die niet zo handig zijn met soldeerbout en tin.

Aegis is een van de bekendste softwarehuizen voor de Amiga en is kwalitatief duidelijk toonaangevend. Met hun nieuwe Modeler-3D maken ze hun reputatie wel waar. Er zijn ook wel andere 3-D programma's zoals Sculpt, maar dat is meer een creatief programma, terwijl Modeler 3D een echt ontwerppakket is.

Modeler 3D van Aegis

Driedimensionaal vervolg op VideoScape 3D

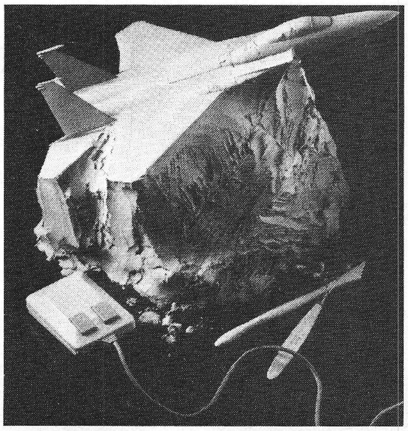
Het hanteren van driedimensionele beelden op een Amiga-scherm is nu nog makkelijker geworden. De minimum-eisen die Aegis vermeldt zijn: 512 k RAM en Workbench versie 1.2 of later. Aanbevolen wordt verder een hard disk en meer RAM.

Roteren, het gebruik van lagen, aktieve polygonen, voorgrond, de normaal, vertex, vlak en nog meer (Engelse) termen komen te pas bij het werken in drie dimensies. Een aantal begrippen is niet meteen in het nederlande te vertalen. De handleiding van Modeler 3D is Engels, maar zeer duidelijk en overzichtelijk. De gebruiker moet overigens wel enige kennis van zaken hebben en ervaring met computer-tekenen.

Stap voor stap kan men vanuit het platte vlak driedimensionaal gaan werken en figuren opbouwen. Eerst eenvoudige, zoals een kubus, later ook meer ingewikkelde en samengestelde figuren, al dan niet met gebogen lijnen. Vervolgens kunnen nog meer bewerkingen worden uitgevoerd, zoals schaal veranderen (zooming), (deel)verplaatsingen, het werken met 'grids', 'templates', projekties, paletten, schaduwen, bewegingen, funkties als 'lathe', 'extrude', 'sculpt 3D' en 'statistics'.

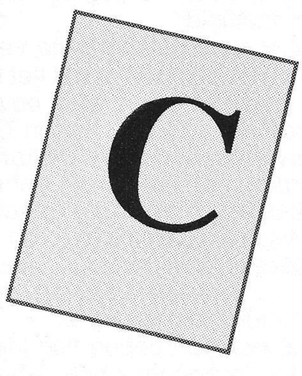
Modeler 3D is een pakket dat de grafische mogelijkheden van de Amiga ten volle weet uit te buiten.

De Nederlandse importeur is Altycos te Zoetermeer, tel. 079 - 510757



Lattice C compiler versie 4.0 is alweer enige maanden op de markt. In dit artikel leggen we Lattice naast concurrent Aztec C. Hiermee hopen we een zo goed mogelijk beeld te geven van de kwaliteiten en mogelijkheden van beide compilers. Ze doen geen van beide voor elkaar onder, ze vullen elkaar eigenlijk aan.

Lattice C Versie 4.0



Voor de eerste indruk gaat men over het algemeen op de plaatjes en handleidingen af. Lattice C scoort hier naar onze mening slechter dan Aztek. Dit vanwege de wat minder solide uitgevallen handleiding. De ruim 400 bladzijden dikke handleiding is een door een ringband bijeengehouden boekwerk. Dit geeft al aan wat er met deze handleiding kan gebeuren.

(Ex)64 bezitters herrineren zich ongetwijfeld de fameuze 'Reference Guide' nog wel. Hoe lang heeft die van u het uitgehouden? Staat nu zeker in een iets solidere map in de boekenkast? Goed, hier scoort Aztek dus een puntje.
Maar, wat ziet ons oog als we Lattice C opstarten? Een heuse programmeer omgeving! Bij het opstarten wordt eerst gevraagd naar datum en tijd. Daarna komt er een bladzijde vol tekst. Na enige tijd lezen wordt het ons duidelijk. Er worden twee utilities opgestart. De een is een zogenaamde 'Screen Saver'. Dit betekent dat het beeldscherm, na twee minuten geen toets- of muisingave, op zwart gaat. Bespaart de monitor weer nietwaar?

Console Handler

Ten tweede wordt er een zogenaamde 'Console Handler' opgestart. Deze wordt overigens pas actief als u een nieuwe CLI start. Wat is nu een console handler? U kunt zich de situatie wel voorstellen. U typt een ellenlange zin in en wat blijkt, er staat een typefoutje in het begin van de regel. Goed, houdt de BACKSPACE toets maar ingedrukt en begin overnieuw. Maar met de console handler is dit niet nodig, even op CURSOR BACK drukken net zo lang tot u op de foute letter staat en verbeteren maar. De console handler beschikt ook over een zogenaamde 'history buffer'. In deze history buffer worden de vorige commando's opgeslagen. Als u nu op CURSOR UP drukt, zullen de voorgaande commando's over uw scherm flitsen. Stop op de commando regel die u weer wilt gebruiken, en opnieuw uitvoeren maar.

De benodigde configuratie

Om met Lattice C te kunnen werken is minimaal een 512 KB Amiga nodig met 2 diskdrives. Dit in tegenstelling tot Aztek C. In hoeverre dit een minpunt is valt te bekijken. De echte Lattice C fans vinden dat het 'veel professioneler staat'. Om 'lekker' te kunnen werken met Lattice C is minimaal een 2.5 MB Amiga met twee diskdrives en een Hard Disk van minimaal 20 MB nodig. Toch mag dit naar onze mening geen bezwaar zijn om met Lattice C te beginnen, immers de diskdrives worden steeds goedkoper en voor veel toepassingen is al een tweede diskdrive nodig.

De handleiding

Zoals we al eerder hebben gezegd, de handleiding is er een van inferieure kwaliteit maar dit beinvloedt de inhoud niet. Het boek is ongeveer 400 bladzijden dik en is geschreven in de Engelse taal. Dit zal voor de meesten onder ons geen bezwaar meer zijn daar Engels de meest gekozen vreemde taal is in het middelbaar onderwijs.
De meeste aandacht wordt geschonken aan de diverse functies die de compiler ondersteunt. Dit vult plusminus 250 bladzijden. De overige hoofdstukken geven uitleg over de Linker, de Compiler, de Errorcodes, etc. Een positief punt is dat er uitleg wordt gegeven over hoe u uw systeem moet opzetten. Dit scheelt weer urenlang knoeien en testen hoe u een fatsoenlijke werkschijf krijgt. Aan dit onderdeel wordt een compleet hoofdstuk besteed.

De schijven

Het pakket komt compleet met vier diskettes. De eerste schijf is de eigenlijke opstartschijf met compiler, linker en assembler. Verder staan er nog een aantal utilities op. Dit zijn een Object Module Disassembler. Hiermee kunt u de object files, dit zijn de welbekende '.o' files, disassembleren oftewel bekijken.
Met de Object Module Librarian kunt u uw eigen library files creëeren. Dit is zeer handig voor het zelf fabriceren van functies.
Diskette 2 is de schijf met de include files, library files en source files. Wat ons onmiddelijk opviel was dat de include files 'gecompressed' waren. Dit verhoogt de compile tijd weer enigszins. Als laatste is op deze schijf dan nog het script aanwezig voor het totaal opstarten van Lattice C. Dit wil zeggen, er is een ASCII bestand, een batch bestand, die de verschillende utilities, de 'Screen Saver' en 'Console Handler', opstart.
Op diskette 3 is weer een include file aanwezig. Hierin staan de niet gecomprimeerde include files. Deze zijn dus gewoon met het TYPE commando te 'listen'. Verder staan op deze schijf, in de directory 'examples', nog diverse voorbeelden.
Als laatste is er dan diskette 4. Deze schijf hoort eigenlijk niet meer bij de programmeer omgeving maar is meer een soort toegift. Er staan kleine routines op zoals 'Avail' en 'Frags'.

Verschillen met versie 3.10

Wat voor verschillen zijn er nu te ontdekken tussen de voorgaande versie 3.10 en de nieuwe 4.0 versie?
Ten eerste, een heel groot verschil, ALINK, de linker die bij versie 3.10 kwam is verdwenen en vervangen door BLINK. Blink vervult zijn taak sneller en maakt kleinere codes, maar hierover straks meer. Op een aantal punten is versie 4.0 zelfs incompatible met v3.xx. Ten eerste zijn de libraries waarmee Blink werkt, niet compatible met Alink. De library voor floating point operaties is zodanig veranderd dat een file persé opnieuw gecompiled moet worden, het afhandelen van zo-

genaamde 'Break Characters' is veranderd. Break Characters zijn de CTRL-C en CTRL-D codes. Verder zijn de INCLUDE files gecomprimeerd, dit ter verhoging van de compile snelheid.
Een van de belangrijkste veranderingen in de compiler is dat het nu mogelijk is van te voren op te geven of het programma in FAST of in CHIP memory moet opereren. Dit kan gebruikt worden om het CHIP geheugen te ontlasten, daar in dat geheugen alle grafische afbeeldingen en alle geluidsdata moeten staan. Hierbij wordt wel verondersteld dat een geheugenuitbreiding aanwezig is.
Er is ook een debug flag bijgekomen zodat de zogenaamde 'label namen' in het compile proces opgeslagen worden. Hierdoor kunt u later met de OMD, Object Module Disassembler, de file opnieuw bekijken.
Veranderingen in de libraries gaan onder andere een stap in de richting van UNIX/XENIX. Er zijn meerdere nieuwe functies bijgekomen die taken vervullen gelijk aan hun naamgenoten onder UNIX/XENIX. De printf() functie is verbeterd.
Als laatste uit de lange lijst veranderingen noemen we de nieuwe Motorola Fast Floating Point libraries. die het net genoemde formaat ondersteunen. Ook het aantal utilities is uitgebreid. We hebben ze eigenlijk al eerder genoemd. In het kort volgen ze hier nog even:

ASM -	Assembler.
BLINK -	Nieuwe linker.
LCOMPACT -	Om header files te comprimeren.
MEMWATCH-	Controleert het laagste deel van het geheugen op errors.
OML -	Een Object Module Librarian.
POPCLI -	Een window utility.
CONSOLE HANDLER -	Voor handiger command line editing.

De compiler

De compiler is een zogenaamde two-pass compiler. Dit betekent dat de compiler uit twee gedeelten bestaat, LC1 en LC2. Gelukkig hoeft u ze niet zelf achter elkaar op te roepen. Hiervoor is al een utility geschreven die dit doet, waarbij u de compiler opties dan op kunt geven. De eerste pass wordt dus door LC1 uitgevoerd. Hierbij wordt de ASCII file die u ingevoerd heeft omgezet naar een zogenaamd QUAD file. Dit Quad file is een tussenliggende vorm. Het is nog geen machinecode. De tweede pass wordt uitgevoerd door LC2. Hierbij wordt de door LC1 aangemaakte Quad file omgezet naar een OBJECT file. Dit Object file is nog niet direkt uit te voeren. het Object file is nog niet relocatable. Dat wil zeggen, het kan nog niet op elke willekeurige plaats in het geheugen geplaatst worden. Zowel bij LC1 als LC2 kunnen meerdere opties gebruikt worden. Het is bijvoorbeeld mogelijk met de Lattice C compiler mogelijk een programma in C te schrijven en deze dan met een optie te compilen, zodat ze op ROM gezet kan worden.

De linker

Zoals we al eerder gezegd hebben, de bij Lattice versie 4.0 geleverde linker is brandnieuw. Dit betekent ook automatisch enige incompatibiliteit. Maar geen getreur, de positieve punten wegen ruimschoots op tegen de negatieve punten. Blink is sneller, maakt kortere codes en heeft meer opties dan Alink. Het gebruik van Blink is identiek aan dat van Alink. Dit wil zeggen dat de commandoregel die u intypte met Alink, grotendeels gelijk is aan wat u met Blink intypt.
Een grote verandering ten opzichte van Alink is dat Blink gebruik kan maken van OVERLAY'S. Bij Overlay's is het mogelijk geheugen, dat in gebruik is geweest door code- of datagedeelten, weer vrij te maken en te gebruiken voor andere codes of data. Dit is vooral voor computers met kleine geheugenruimten of computers die door hun inefficiënte geheugengebruik weinig geheugen overhouden.
Dit proces heeft een nadeel en een voordeel. Het nadeel is dat Blink iets trager wordt door het laden. Het voordeel is dat het geheugengebruik door Blink veel efficiënter wordt.

De assembler

Deze assembler genaamd ASM is een volwaardige mnemonics assembler die een grote set van assembler-directives, lees assembler stuurcodes, ondersteund. Met Asm kunt u gedeelten machinecode voor-compilen om ze later te gebruiken in uw C programma. Het is ook mogelijk stukken machine code apart te schrijven, met Asm te assembleren en vervolgens te linken. Daarna heeft u een stuk uitvoerbare code die u in mnemonics geschreven hebt.
Het blijft toch jammer dat de mnemonics nog niet direct in het C programma te integeren zijn. Dit blijft dus nog een van de sterke punten van Aztek C.

Vergelijkende test

Nu de 'big key question', welke is beter? De beste oplossing leek ons een klein programmaatje te schrijven, om deze vervolgens met beide compilers te testen op grootte, compile- en link tijd en op uitvoersnelheid. Het programma op zich voerde niet zoveel uit. Het opende de timerport en telde van 0 tot 1000000. Het programma stond in beide gevallen op RAM disk. De libraries moesten in beide gevallen van de schijf gehaald worden. De resultaten waren alsvolgt:

	Aztek C	Lattice C
Compilen:	16.5 sec.	26.5 sec.
Linken:	18.4 sec.	19.0 sec.
File grootte:	5644 bytes	8544 bytes
Uitvoertijd:	8.513 sec.	9.980 sec.

Het compilen bij Aztek C bestond uit de compiler aanroepen (cc) en de assembler (as). Lattice C riep LC1 en LC2 aan tijdens de compile fase. Uit deze test komt Aztek C als overwinnaar uit de strijd. We vonden dit bewijs wel een beetje eenzijdig en hebben daarom het programma een beetje ingewikkelder gemaakt. We hebben de rekenvaardigheden van de compilers getest.
De som b= a * 50 / (53 % 4) + a is een miljoen maal berekend. Even als opmerking, het was een integer bewerking, geen floating point. De volgende tabel doet van deze test verslag:

	Aztek C	Lattice C
Compilen:	16.8 sec.	29.0 sec.
Linken:	18.6 sec.	19.2 sec.
File grootte:	5664 bytes	8588 bytes
Uitvoertijd:	62.706 sec.	115.982 s.

Ook uit deze test komt Aztek C weer als winnaar tevoorschijn. Hebben we niet ergens gehoord dat Lattice C versie 5.0 al op stapel staat? Goed, uit de test blijkt ook wel dat het nodig is. Op de linker na is Lattice C slechter dan Aztek C. Blink echter doet het verbazend goed ten opzichte van de Aztek linker. Hierin hoeven we waarschijnlijk dan ook geen verbeteringen te verwachten, deze is gewoon al goed.
Ten opzichte van Aztek C moet Lattice C helaas nog steeds het onderspit delven. Eigenlijk op alle fronten is de concurrent beter. Toch blijft Lattice C een goed stuk programmeer werk. Ze blijft zeer compatible met haar gelijknamige tegenhangers op de PC's. De Lattice versie 4.0 compiler kost *ƒ* 649,-. Voor dit geld krijgt u een 400 bladzijden dikke handleiding en vier schijven.

Inl: Het Computercollectief, Amstel 312, 1017 AP, Amsterdam.

Soft Wir War

Bert Tier & Wouter Meijer

